1 Programming

Table of contents

© 2002-2018 Jørgen Sørensen
Web site: http://www.jososoft.dk/yamaha
E-mail: js@jososoft.dk

1.1 CASM Section Format 1

The CASM is the first Yamaha-chunk in a style file following the MIDI data.

The CASM section holds the Ctab settings and some other settings of the style. Only some of these settings can be done at the keyboard.

In December 1999 I reverse-engineered the CASM. The original format was published March 2000.

By the arrival of the first Tyros keyboard the CASM section format was altered slightly.

These changes were reverse engineered in February 2003. The revised format was published October 25th 2004.

The CASM format was changed radically with the release of Tyros 3 in November 2008. This new format has been applied to later keyboards.

Check CASM Editor software program at http://www.jososoft.dk/yamaha/software/casmedit/index.htm

In the following x represents a byte.

No.	CASM data	Function	Hex value	Comment
00	CASM	CASM Marker	43 41 53 4d	Beginning of CASM Section
01	xxxx	CASM Length		Length of the entire CASM section
02	CSEG	Section Marker	43 53 45 47	Beginning of CSEG Section within CASM. A CSEG Section holds information about style parts using equal settings.
03	xxxx	Section Length		Length of the CSEG section
04	Sdec	Parts Marker	53 64 65 63	Beginning of Sdec part within CSEG Section
05	xxxx	Parts Length		Length of the Sdec part
06	Main A,Main B etc.	Style parts		Style part names of the styles with this setting. Names are separated with commas, but no comma at the end of the part name string.
07	Ctab	Channel Marker	43 74 61 62	Beginning of Ctab string
08	xxxx	Channel Length		Length of the Ctab string
09	x	Source Channel	00 - 0F	The source channel in the MIDI part of the style file which holds note information. Valid values are 00 - 0F (= channel 1 - 16).
10	xxxxxxxx	Voice Name		Voice name. Always 8 bytes. The voice can be called any name.
11	x	Destination Channel	08 - 0F	Source channel data must be remapped to a valid destination channel. Valid values are 08 - 0F (= channel 9 - 16).
12	x	Editable	00 - 01	01 Channel Read Only - 00 Channel Editable
13	x	Note Mute	00 - 0F 1. digit 2. digit	Notes to play (0 = not play) 1.digit = 0 (always)
14	x	Note Mute	00 - FF 3. digit 4. digit	

```
2.digit = F E D C B A 9 8 7 6 5 4 3 2 1 0
     B    1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
     Bb   1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
     A    1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
     G#   1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0

3.digit = F E D C B A 9 8 7 6 5 4 3 2 1 0
     G    1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
     F#   1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
     F    1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
     E    1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
```

```
4.digit = F E D C B A 9 8 7 6 5 4 3 2 1 0
     Eb   1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
     D    1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
     C#   1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
     C    1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
```

Example: 0E A3 means that C C# F G A Bb and B plays. When other notes are pressed the accompaniment is muted.

15	x	Chord Mute	00 - 07 1. digit 2. digit	Chords to play (0 = not play)
16	x	Chord Mute	00 - FF 3. digit 4. digit	
17	x	Chord Mute	00 - FF 5. digit 6. digit	
18	x	Chord Mute	00 - FF 7. digit 8. digit	
19	x	Chord Mute	00 - FF 9. digit 10. digit	

```
1.digit = 0 (always)

2.digit = 3 2 1 0      add 4 if auto start
  1+2+5   1 1 0 0
  sus4    1 0 1 0

3.digit = F E D C B A 9 8 7 6 5 4 3 2 1 0
  1+5     1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
  1+8     1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
  7aug    1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
  M7aug   1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0

4.digit = F E D C B A 9 8 7 6 5 4 3 2 1 0
  7(#9)   1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
  7(b13)  1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
  7(b9)   1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
  7(13)   1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0

5.digit = F E D C B A 9 8 7 6 5 4 3 2 1 0
  7#11    1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
  7(9)    1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
  7b5     1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
  7sus4   1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0

6.digit = F E D C B A 9 8 7 6 5 4 3 2 1 0
  7       1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
  dim7    1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
  dim     1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
  m7M(9)  1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0

7.digit = F E D C B A 9 8 7 6 5 4 3 2 1 0
  mM7     1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
  m7(11)  1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
  m7(9)   1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
  m(9)    1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0

8.digit = F E D C B A 9 8 7 6 5 4 3 2 1 0
  m7b5    1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
  min7    1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
  min6    1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
  min     1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0

9.digit = F E D C B A 9 8 7 6 5 4 3 2 1 0
  aug     1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
  6(9)    1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
  M7(9)   1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
   (9)    1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0

10.digit= F E D C B A 9 8 7 6 5 4 3 2 1 0
  M7#11   1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
  Maj7    1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
  Maj6    1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
  Maj     1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
```

Example: 03 FF EF FF F8 means that when playing a Maj chord or a 7sus4 chord the accompaniment is muted.

20	x	Source Chord	00 - 0B	These settings determine the original key of the source pattern (i.e. the key used when recording the pattern). The default, CM7 (the source root is "C" and the source chord type is "M7"), is automatically selected whenever the preset data is deleted prior to recording a new style, regardless of the source root and chord included in the preset data. Valid Chord Types are: Maj, Maj6, Maj7, M7#11, Madd9, M7(9), M6(9), aug, m, m6, m7, m7b5, m(9), m7(9), m7(11), mM7, mM7(9), dim, dim7, 7, 7sus, 7b5, 7(9), 7(#11), 7(13), 7(b9), 7(b13), 7(#9), M7aug, 7aug, 1+8, 1+5, sus4, 1+2+5
21	x	Chord Type	00 - 21	
22	x	Note Transposition Rule	00 - 01	NTR specify the transposition rule to be used by the transposition table. Two settings are available: ROOT TRANS (00): When transposed the pitch relationship between notes is maintained. For example, the notes C3, E3, and G3 in the key of C will become F3, A3, and C4 when transposed to F. Use this setting for parts that contain melodic lines. ROOT FIXED (01): The note is kept as close as possible to the previous note range. For example, the notes C3, E3, and G3 in the key of C will become C3, F3, and A3 when transposed to F. Use this setting for chordal parts.
23	x	Note Transposition Table	00 - 05	NTT specify the note transposition table to be used for source pattern transposition. The following table types are available. Note that to obtain the settings marked with *, the NTT setting in the Cntt section (line 32) must be set too. BYPASS (00): No transposition. MELODY (01): Suitable for melody line transposition. Use for melody parts such as PHRASE 1 and PHRASE 2. CHORD (02): Suitable for chord transposition. Use for the CHORD 1 and CHORD 2 parts when they contain piano or guitar-like chordal parts. BASS (03): Suitable for bass line transposition. This table is basically similar to the MELODY table, but recognizes "on-bass" chords allowed in the FINGERED 2 fingering mode. Use primarily for bass lines. MELODIC MINOR (04): This table lowers the third scale degree by a semitone when changing from a major to a minor chord, or raises the minor third scale degree a semitone when changing from a minor to a major chord. Other notes are not changed. *MELODIC MINOR 5th Variation (04) HARMONIC MINOR (05): This table lowers the third and sixth scale degrees by a semitone when changing from a major to a minor chord, or raises the minor third and flatted sixth scale degrees a semitone when changing from a minor to a major chord. Other notes are not changed. *HARMONIC MINOR 5th Variation (05) *NATURAL MINOR. If BassOff (04). If BassOn (05) *NATURAL MINOR 5th Variation (04) *DORIAN (04) *DORIAN 5th Variation (04) NOTE When NTR (above) is set to ROOT FIXED and NTT (also above) is set to BYPASS, the SOURCE ROOT

				and SOURCE CHORD parameter names change to PLAY ROOT and PLAY CHORD. In this case it is possible to change chords and hear how the results sound for all parts. If "P" or "PRESET" appears for the SOURCE ROOT, SOURCE CHORD, NTR, or NTT parameter, the preset data uses special settings.
24	x	High key	00 - 0B	HIGH KEY specify the upper root limit. Chords with a root higher than the specified limit will be played in the octave immediately below the high-key limit. This setting is effective only when the NTR parameter (above) is set to ROOT TRANS. Example: When HIGH KEY = F. Root Motion: C C# D F F# Notes Produced: C3-E3-G3 / C#3-F3-G#3 / D3-F#3-A3 / F3-A3-C4 / F#2-A#2-C#3
25	x	Note Low Limit	00 - 7F	NOTE LIMIT LOW and HIGH specify the low and high note limits for all notes in the specified part. Notes outside this range are transposed to the nearest octave within the range. Example: When LOW = C3 and HIGH = D4 Root Motion: C C# D# Notes Produced: E3-G3-C4 / F3-G#3-C#4 / D#3-G3-A#3
26	x	Note High Limit	00 - 7F	
27	x	Retrigger Rule	00 - 05	RTR (Retrigger Rule) specify how notes held through chord changes will be handled. 6 settings are available: STOP (00): The note is stopped, and resumes sounding from the next note data. PITCH SHIFT (01): The pitch of the note will bend without attack to match the type of the new chord. PITCH SHIFT TO ROOT (02): The pitch of the note will bend without attack to match the root of the new chord. RETRIGGER (03): The note is retriggered with attack at a new pitch matching the new chord type. RETRIGGER TO ROOT (04): The note is retriggered with attack at a new pitch matching the new chord root. NOTE GENERATOR (05): This setting will only be available if programmed in the original style. A designated note is produced with designated pitch, length, and velocity matching the new chord.
28	x	End Marker	00	End Marker for this channel

... and over from line 07 for each **channel** in the CSEG section.
When all channels are set, continue from line 29 with all channels with defined values in line 32.

29	Cntt	New NTT	43 6E 74 74	Beginning of Cntt string
30	xxxx	Cntt Length	00 00 00 02	Length of the Cntt string
31	x	Source Channel	00 - 0F	The source channel. Same value as in line 09.
32	x	Note Transposition Table	see next column	NTT specify the note transposition table to be used for source pattern transposition. 11 table types are available:

Type	Bass Off	Bass On
BYPASS	not def.	80
MELODY + BASS	not def.	81

CHORD	not def.	82
MELODIC MINOR	not def.	83
MELODIC MINOR 5th VAR	04	84
HARMONIC MINOR	not def.	85
HARMONIC MINOR 5th VAR	06	86
NATURAL MINOR	07	87
NATURAL MINOR 5th VAR	08	88
DORIAN	09	89
DORIAN 5th VAR	0A	8A

... and now over from line 29 for each **channel** with defined values in line 32 in the CSEG section

... and now over from line 02 for each **CSEG** section in CASM

1.2 CASM Section Format 2

The CASM is the first Yamaha-chunk in a style file following the MIDI data.

The CASM section holds the Ctab settings and some other settings of the style. Only some of these settings can be done at the keyboard.

In December 1999 I reverse-engineered the original CASM. This was published March 2000.

By the arrival of the first Tyros keyboard the original CASM section format was altered slightly.

These changes were reverse engineered in February 2003. The revised format was published October 25th 2004.

The CASM format was changed radically with the release of Tyros 3 in November 2008. The new format has been applied to later keyboards.

In the following x represents a byte.

No.	CASM data	Function	Hex value	Comment
00	CASM	CASM Marker	43 41 53 4d	Beginning of CASM Section
01	xxxx	CASM Length		Length of the entire CASM section
02	CSEG	Section Marker	43 53 45 47	Beginning of CSEG Section within CASM. A CSEG Section holds information about style parts using equal settings.
03	xxxx	Section Length		Length of the CSEG section
04	Sdec	Parts Marker	53 64 65 63	Beginning of Sdec part within CSEG Section
05	xxxx	Parts Length		Length of the Sdec part
06	Main A,Main B etc.	Style parts		Style part names of the styles with this setting. Names are separated with commas, but no comma at the end of the part name string.
07	Ctab	Channel Marker	43 74 61 62	Beginning of Ctab string
08	xxxx	Channel Length		Length of the Ctab string
09	x	Source Channel	00 - 0F	The source channel in the MIDI part of the style file which holds note information. Valid values are 00 - 0F (= channel 1 - 16).
10	xxxxxxxx	Voice Name		Voice name. Always 8 bytes. The voice can be called any name.
11	x	Destination Channel	08 - 0F	Source channel data must be remapped to a valid destination channel. Valid values are 08 - 0F (=

				channel 9 - 16).
12	x	Editable	00 - 01	01 Channel Read Only - 00 Channel Editable
13	x	Note Mute	00 - 0F 1. digit 2. digit	Notes to play (0 = not play)
14	x	Note Mute	00 - FF 3. digit 4. digit	

```
1.digit = 0 (always)

2.digit = F E D C B A 9 8 7 6 5 4 3 2 1 0
     B    1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
     Bb   1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
     A    1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
     G#   1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0

3.digit = F E D C B A 9 8 7 6 5 4 3 2 1 0
     G    1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
     F#   1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
     F    1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
     E    1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0

4.digit = F E D C B A 9 8 7 6 5 4 3 2 1 0
     Eb   1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
     D    1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
     C#   1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
     C    1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
```

Example: 0E A3 means that C C# F G A Bb and B plays. When other notes are pressed the accompaniment is muted.

15	x	Chord Mute	00 - 07 1. digit 2. digit	Chords to play (0 = not play)
16	x	Chord Mute	00 - FF 3. digit 4. digit	
17	x	Chord Mute	00 - FF 5. digit 6. digit	
18	x	Chord Mute	00 - FF 7. digit 8. digit	
19	x	Chord Mute	00 - FF 9. digit 10. digit	

```
1.digit = 0 (always)

2.digit = 3 2 1 0       add 4 if auto start
  1+2+5   1 1 0 0
  sus4    1 0 1 0

3.digit = F E D C B A 9 8 7 6 5 4 3 2 1 0
  1+5     1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
  1+8     1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
  7aug    1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
  M7aug   1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0

4.digit = F E D C B A 9 8 7 6 5 4 3 2 1 0
  7(#9)   1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
  7(b13)  1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
  7(b9)   1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
  7(13)   1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0

5.digit = F E D C B A 9 8 7 6 5 4 3 2 1 0
  7#11    1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
  7(9)    1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
  7b5     1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
  7sus4   1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0

6.digit = F E D C B A 9 8 7 6 5 4 3 2 1 0
  7       1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
  dim7    1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
  dim     1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
  m7M(9)  1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0

7.digit = F E D C B A 9 8 7 6 5 4 3 2 1 0
  mM7     1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
  m7(11)  1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
  m7(9)   1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
  m(9)    1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0

8.digit = F E D C B A 9 8 7 6 5 4 3 2 1 0
  m7b5    1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
  min7    1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
  min6     1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
  min      1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0

9.digit =  F E D C B A 9 8 7 6 5 4 3 2 1 0
  aug      1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
  6(9)     1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
  M7(9)    1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
  (9)      1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0

10.digit=  F E D C B A 9 8 7 6 5 4 3 2 1 0
  M7#11    1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
  Maj7     1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
  Maj6     1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
  Maj      1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
```

Example: 03 FF EF FF F8 means that when playing a Maj chord or a 7sus4 chord the accompaniment is muted.

20	x	Source Chord	00 - 0B	These settings determine the original key of the source pattern (i.e. the key used when recording the pattern). The default, CM7 (the source root is "C" and the source chord type is "M7"), is automatically selected whenever the preset data is deleted prior to recording a new style, regardless of the source root and chord included in the preset data. Valid Chord Types are: Maj, Maj6, Maj7, M7#11, Madd9, M7(9), M6(9), aug, m, m6, m7, m7b5, m(9), m7(9), m7(11), mM7, mM7(9), dim, dim7, 7, 7sus, 7b5, 7(9), 7(#11), 7(13), 7(b9), 7(b13), 7(#9), M7aug, 7aug, 1+8, 1+5, sus4, 1+2+5
21	x	Chord Type	00 - 21	
22	x	Low / middle limit	00 - 7F	Specifies the lowest MIDI note value which is part of the "middle note section". All notes below this note belong to the "low notes section". If the value of this byte is 0, then the data in the "low notes section" is not used.
23	x	Middle / high limit	00 - 7F	Specifies the highest MIDI note value which is part of the "middle note section". All notes above this note belong to the "high notes section". If the value of this byte is 7F, then the data in the "high notes section" is not used.

The following 6 lines are repeated for low, middle and high notes sections.

24	x	Note Transposition Rule	00 - 02	NTR specify the transposition rule to be used by the transposition table. Three settings are available: ROOT TRANS (00): When transposed the pitch relationship between notes is maintained. For example, the notes C3, E3, and G3 in the key of C will become F3, A3, and C4 when transposed to F. Use this setting for parts that contain melodic lines. ROOT FIXED (01): The note is kept as close as possible to the previous note range. For example, the notes C3, E3, and G3 in the key of C will become C3, F3, and A3 when transposed to F. Use this setting for chordal parts. GUITAR (02): This is exclusively for transposing guitar accompaniment. Notes are transposed to approximate the chords played with natural guitar fingering.
25	x	Note Transposition Table	Bass "off": 00 - 0A Bass "on":	NTT specify the note transposition table to be used for source pattern transposition. The following table types are available. BYPASS (00 - 80): No transposition.

			80 - 8A	MELODY (01 - 81): Suitable for melody line transposition. Use for melody parts such as PHRASE 1 and PHRASE 2. CHORD (02 - 82): Suitable for chord transposition. Use for the CHORD 1 and CHORD 2 parts when they contain piano or guitar-like chordal parts. MELODIC MINOR (03 - 83): This table lowers the third scale degree by a semitone when changing from a major to a minor chord, or raises the minor third scale degree a semitone when changing from a minor to a major chord. Other notes are not changed. MELODIC MINOR 5th Variation (04 - 84) HARMONIC MINOR (05 - 85): This table lowers the third and sixth scale degrees by a semitone when changing from a major to a minor chord, or raises the minor third and flatted sixth scale degrees a semitone when changing from a minor to a major chord. Other notes are not changed. HARMONIC MINOR 5th Variation (06 - 86) NATURAL MINOR (07 - 87) NATURAL MINOR 5th Variation (08 - 88) DORIAN (09 -89) DORIAN 5th Variation (0A - 8A) ALL-PURPOSE (00 - 80): This table covers both strummed- and arpeggio-played sound. STROKE (01 - 81): Suitable for stroke-played sound of the guitar. Some notes may sound as if it is muted. This is normal condition when the chord is played on guitar by stroke. ARPEGGIO (02 - 82): Suitable for arpeggio-played sound of the guitar. Using this table, four notes arpeggio sounds most beautiful. NOTES If Bass is "off" values 00 - 0A are used, else if Bass is "on" values 80 - 8A are used. These settings are in the keyboard fingering area. ALL-PURPOSE; STROKE and ARPEGGIO is only available when NTR (above) is set to GUITAR.
26	x	High key	00 - 0B	HIGH KEY specify the upper root limit. Chords with a root higher than the specified limit will be played in the octave immediately below the high-key limit. This setting is effective only when the NTR parameter (above) is set to ROOT TRANS. Example: When HIGH KEY = F. Root Motion: C C# D F F# Notes Produced: C3-E3-G3 / C#3-F3-G#3 / D3-F#3-A3 / F3-A3-C4 / F#2-A#2-C#3
27	x	Note Low Limit	00 - 7F	NOTE LIMIT LOW and HIGH specify the low and high note limits for all notes in the specified part. Notes outside this range are transposed to the nearest octave within the range. Example: When LOW = C3 and HIGH = D4 Root Motion: C C# D# Notes Produced: E3-G3-C4 / F3-G#3-C#4 / D#3-G3-A#3
28	x	Note High Limit	00 - 7F	
29	x	Retrigger Rule	00 - 05	RTR (Retrigger Rule) specify how notes held through chord changes will be handled. 6 settings are available: STOP (00): The note is stopped, and resumes sounding from the next note data.

				PITCH SHIFT (01): The pitch of the note will bend without attack to match the type of the new chord. PITCH SHIFT TO ROOT (02): The pitch of the note will bend without attack to match the root of the new chord. RETRIGGER (03): The note is retriggered with attack at a new pitch matching the new chord type. RETRIGGER TO ROOT (04): The note is retriggered with attack at a new pitch matching the new chord root. NOTE GENERATOR (05): This setting will only be available if programmed in the original style. A designated note is produced with designated pitch, length, and velocity matching the new chord.

... and over from line 24 - 29 for **middle** notes section; and once more for **high** notes section.

30*)	x		00 or 80	Default value is 00. If value is 80 there is an extra break voice (like a Crash Cymbal in drum channels) for **non-drum channels**, when playing the 3- or 4-finger break. The extra break drum voice will sound at time 0 within the break measure.
31*)	x		00 or 01	Default value is 00. If value is 01, then the channel is always a drum channel. In this case lines 33, 34 and 35 have non-00 values and line 30 is always 00.
32*)	x		00	Always 00.
33*)	x		00 or 18	Default value is 00. If value is 18, then the channel is always a drum channel. In this case line 31 is 01 and line 34 and 35 has non-00 values.
34*)	x	Instrument	23 - 50 or 80	Default value is 80. If line 33 is 18 (= drum channel), then value is in interval 23 - 50. The actual value is the General MIDI Percussion Key Number.
35*)	x	Volume	00 - 7F	Default value is 00. If line 33 is 18 (= drum channel), then actual value is the volume of the Percussion Instrument, defined in the line above.
36	x	End Marker	00	End Marker for this channel

... and over from line 07 for each channel in the CSEG section.

... and now over from line 02 for each CSEG section in CASM

*)
The findings in these lines are still to be verified. In rare cases other values might be present in lines 30, 34 and 35.

This section is from "Style Files - Introduction and Details" by Peter Wierzba and Michael P. Bedesem. The entire document is found at http://www.wierzba.homepage.t-online.de/stylefiles.htm A must read for style programmers!

If NTR is "Guitar" the following apply

In contrast to other NTRs there is no harmonic relation between source and target notes. Each source note is mapped to one of the guitar strings. The pitch or harmonic function will be irrelevant.

The mapping of source notes to guitar strings is as follows:
B -> 1st string (high E)
A -> 2nd string (B)
G -> 3rd string (G)
F -> 4th string (D)
E -> 5th string (A)

D -> 6th string (low E)
C# -> a quint above/below
C -> root note

That means you can control exactly which of the six strings should sound at what time.

C and C# will be mapped to the root of an on-bass chord, if parameter BASS is set to on.

It is recommended not to use C, C#, D and E at the same time.

If source notes will be moved by an octave this does not mean that the chord will sound an octave lower/higher.

Rather you can control which chord position on the fretboard will be used:
C2 - B2 -> 1st position
C3 - B3 -> 2nd position
C4 - B4 -> 3rd position
C5 - B5 -> 4th position

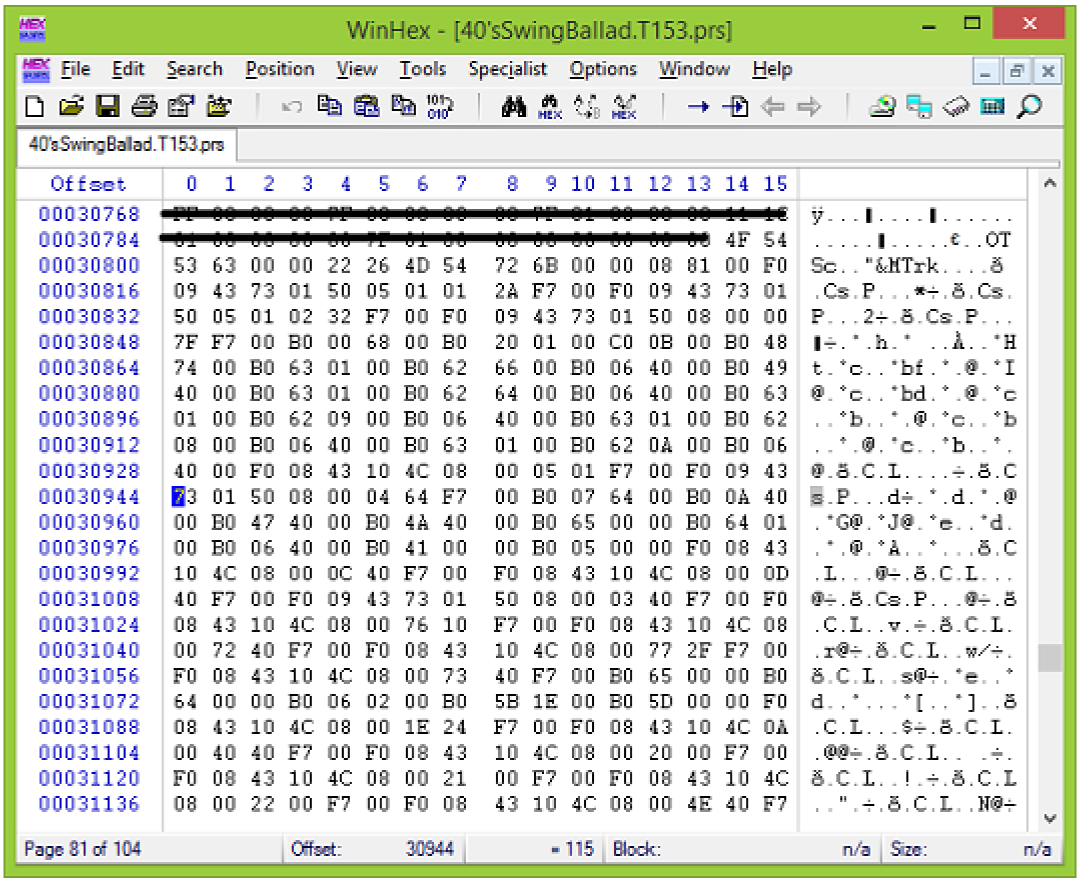
As a consequence of the above information, the MIDI channels in the style file having the "Guitar" setting in NTR must be prepared accordingly.
This also means that the style file will not sound right when/if switching between "Guitar" and one of the other NTR settings without editing the MIDI notes.

1.3 OTS Section Format

The OTS is the second Yamaha-chunk in a style file following the CASM data.

OTS Editor software program at http://www.jososoft.dk/yamaha/software/otseditor/index.htm

In a hexadecimal editor the beginning of the OTS section might look like this:


WinHex - [40'sSwingBallad.T153.prs]
File Edit Search Position View Tools Specialist Options Window Help
40'sSwingBallad.T153.prs

```
  Offset    0  1  2  3  4  5  6  7   8  9 10 11 12 13 14 15
00030768   ~~FF 00 00 00 FF 00 00 00  00 FF 01 00 00 00 11 1E~~  ÿ...▮....▮......
00030784   ~~01 00 00 00 00 7F 01 00  00 00 00 00 00 00~~ 4F 54  .....▮.....€..OT
00030800   53 63 00 00 22 26 4D 54  72 6B 00 00 08 81 00 F0  Sc.."&MTrk....ð
00030816   09 43 73 01 50 05 01 01  2A F7 00 F0 09 43 73 01  .Cs.P...*÷.ð.Cs.
00030832   50 05 01 02 32 F7 00 F0  09 43 73 01 50 08 00 00  P...2÷.ð.Cs.P...
00030848   7F F7 00 B0 00 68 00 B0  20 01 00 C0 0B 00 B0 48  ▮÷.°.h.° ..À..°H
00030864   74 00 B0 63 01 00 B0 62  66 00 B0 06 40 00 B0 49  t.°c..°bf.°.@.°I
00030880   40 00 B0 63 01 00 B0 62  64 00 B0 06 40 00 B0 63  @.°c..°bd.°.@.°c
00030896   01 00 B0 62 09 00 B0 06  40 00 B0 63 01 00 B0 62  ..°b..°.@.°c..°b
00030912   08 00 B0 06 40 00 B0 63  01 00 B0 62 0A 00 B0 06  ..°.@.°c..°b..°.
00030928   40 00 F0 08 43 10 4C 08  00 05 01 F7 00 F0 09 43  @.ð.C.L....÷.ð.C
00030944   73 01 50 08 00 04 64 F7  00 B0 07 64 00 B0 0A 40  s.P...d÷.°.d.°.@
00030960   00 B0 47 40 00 B0 4A 40  00 B0 65 00 00 B0 64 01  .°G@.°J@.°e..°d.
00030976   00 B0 06 40 00 B0 41 00  00 B0 05 00 00 F0 08 43  .°.@.°A..°...ð.C
00030992   10 4C 08 00 0C 40 F7 00  F0 08 43 10 4C 08 00 0D  .L...@÷.ð.C.L...
00031008   40 F7 00 F0 09 43 73 01  50 08 00 03 40 F7 00 F0  @÷.ð.Cs.P...@÷.ð
00031024   08 43 10 4C 08 00 76 10  F7 00 F0 08 43 10 4C 08  .C.L..v.÷.ð.C.L.
00031040   00 72 40 F7 00 F0 08 43  10 4C 08 00 77 2F F7 00  .r@÷.ð.C.L..w/÷.
00031056   F0 08 43 10 4C 08 00 73  40 F7 00 B0 65 00 00 B0  ð.C.L..s@÷.°e..°
00031072   64 00 00 B0 06 02 00 B0  5B 1E 00 B0 5D 00 00 F0  d..°...°[..°]..ð
00031088   08 43 10 4C 08 00 1E 24  F7 00 F0 08 43 10 4C 0A  .C.L...$÷.ð.C.L.
00031104   00 40 40 F7 00 F0 08 43  10 4C 08 00 20 00 F7 00  .@@÷.ð.C.L.. .÷.
00031120   F0 08 43 10 4C 08 00 21  00 F7 00 F0 08 43 10 4C  ð.C.L..!.÷.ð.C.L
00031136   08 00 22 00 F7 00 F0 08  43 10 4C 08 00 4E 40 F7  ..".÷.ð.C.L..N@÷
```

```
Page 81 of 104    Offset:    30944    = 115    Block:    n/a    Size:    n/a
```

The OTS chunk identifier "OTSc" and then 4 bytes giving the length of the entire section, not including the 4 identifier bytes and the 4 length bytes.

Next comes a "MTrk" identifier. This is a MIDI track identifier; and later on will follow 3 more "MTrk" identifiers.

The first MIDI Track holds the "OTS 1" information; the second MIDI Track holds the "OTS 2" information; and so on.

This means that the entire OTS section is so to speak a "4 Track MIDI File with an abnormal header".

In the MIDI Tracks we will find some "normal" MIDI events and a lot of "normal" SysEx messages defining the voices in the OTS set-up.

These "normalities" are found in the keyboard data sheet. SysEx of this form: "F0 43 10 4C xx xx xx yy F7" are normal.

After stripping a OTS section for these "normalities" there will remain some OTS specific data, which all are SysEx messages.

These messages have this form: "F0 43 73 01 5z xx xx xx yy .. yy F7", where
• "z" is 0 or 1
• "xx xx xx" is address. The second xx block is 00 -> 03 = channel (1 -> 4) - Right 1; Right 2; Right 3 and Left.
• "yy .. yy" are actual data value - one or more bytes

Notice:
The information in this page is not complete; and errors might occur.
Yamaha never tells us about their file formats!
Please contact me if you have information about the "?" fields in the table.

Section	**SysEx message**	**Data Byte Value(s)**	**Function**
Header	F0 43 73 01 50 05 01 01 2A F7		?
	F0 43 73 01 50 05 01 02 32 F7		?
Part Right 1	F0 43 73 01 50 08 00 00 yy F7	00 = 0 or 7F = 127	part off/on
	F0 43 73 01 50 08 00 04 yy F7	00 = 0 to 7F = 127	voice set volume *)
	F0 43 73 01 50 08 00 03 yy F7	**See table below**	Octave Set *)
Part Right 2	F0 43 73 01 50 08 01 00 yy F7	00 = 0 or 7F = 127	part off/on
	F0 43 73 01 50 08 01 04 yy F7	00 = 0 to 7F = 127	voice set volume *)
	F0 43 73 01 50 08 01 03 yy F7	**See table below**	Octave set *)
Part Right 3	F0 43 73 01 50 08 02 00 yy F7	00 = 0 or 7F = 127	part off/on
	F0 43 73 01 50 08 02 04 yy F7	00 = 0 to 7F = 127	voice set volume *)
	F0 43 73 01 50 08 02 03 yy F7	**See table below**	Octave set *)
Part Left	F0 43 73 01 50 08 03 00 yy F7	00 = 0 or 7F = 127	part off/on
	F0 43 73 01 50 08 03 04 yy F7	00 = 0 to 7F = 127	voice set volume *)
	F0 43 73 01 50 08 03 03 yy F7	**See table below**	Octave set *)
Part Right 1	F0 43 73 01 51 08 00 11 yy yy yy F7	Check Manual	DSP Variation Parameter *)
	F0 43 73 01 50 08 00 08 yy F7	00 = 0 or 7F = 127	DSP off/on
	F0 43 73 01 50 08 00 01 yy F7	00 = 0 or 7F = 127	DSP Variation off/on *)
Part Right 2	F0 43 73 01 51 08 01 11 yy yy yy F7	Check Manual	DSP Variation Parameter *)
	F0 43 73 01 50 08 01 08 yy F7	00 = 0 or 7F = 127	DSP off/on
	F0 43 73 01 50 08 01 01 yy F7	00 = 0 or 7F = 127	DSP Variation off/on *)
Part Right 3	F0 43 73 01 51 08 02 11 yy yy yy F7	Check Manual	DSP Variation Parameter *)
	F0 43 73 01 50 08 02 08 yy F7	00 = 0 or 7F = 127	DSP off/on
	F0 43 73 01 50 08 02 01 yy F7	00 = 0 or 7F = 127	DSP Variation off/on *)
Part Left	F0 43 73 01 51 08 03 11 yy yy yy F7	Check Manual	DSP Variation Parameter *)
	F0 43 73 01 50 08 03 08 yy F7	00 = 0 or 7F = 127	DSP off/on

	F0 43 73 01 50 08 03 01 yy F7	00 = 0 or 7F = 127	DSP Variation off/on *)
	F0 43 73 01 50 08 03 02 yy F7	00 = 0 or 7F = 127	Left Hold off/on *)
Harmony	F0 43 73 01 50 04 00 00 yy F7	00 = 0 or F7 = 127	off/on
	F0 43 73 01 51 04 00 00 yy yy yy F7	**See table below**	type + speed
	F0 43 73 01 50 04 00 05 yy F7	00 = 0 to 7F = 127	volume
	F0 43 73 01 50 04 00 02 yy F7	**See table below**	assign
	F0 43 73 01 50 04 00 03 yy F7	00 = 0 or FF = 255	ch note off/on
	F0 43 73 01 50 04 00 04 yy F7	00 = 0 to 7F = 127	touch limit
Multi Pad	F0 43 73 01 51 07 00 00 yy yy yy F7	Find values later in this document	number
	F0 43 73 01 50 07 00 00 yy F7	00 = 0 to 7F = 127	volume
Manufacturer Specific message (format unknown)		260 Bytes	?

*) Setting of this parameter and all settings marked "?" are not yet implemented in the OTS Editor software.

Octave Value	**Harmony Data**		

3E = -2
3F = -1
40 = 0
41 = 1
42 = 2
	Types
	Standard Duet = 0
	Standard Trio = 1
	Full Chord = 2
	Rock Duet = 0 + 256
	Country Duet = 0 + 256 * 2
	Country Trio =1 + 256
	Block = 3
	4-Way Close1 = 4
	4-Way Close2 = 4 + 256
	4-Way Open = 5
	1+5 = 6
	Octave = 7
	Strum = 8
	Multi Assign = 9
	Echo = 10
	Tremolo = 11
	Trill = 12
		Speed Values
		Echo type:
		4, 6, 8 and 12
		Tremolo and Trill types:
		8, 12, 16 and 32
		
		Type + Speed Data Bytes
		Byte 1 = always 2
		
		Echo, Tremolo and Trill types:
		Byte 2 = type
		Byte 3 = speed index (0 to 3)
		
		Other Harmony types:
		Byte 2 = type%256
		Byte 3 = type/256
		
		If byte 2 and byte 3 = 0A 00
		Type = Echo
		Speed = 4 (index 0)
			Assign
			Auto = 16
			Multi = 17
			R1 = 0
			R2 = 1
			R3 = 2

In PSR A3000; PSR S970; PSR S770 and PSR S670 there has been added 166 additional Harmony/Arpeggio types.
These has been added to the OTS Editor software at http://www.jososoft.dk/yamaha/software/otseditor/index.htm from version 2.37.

1.4 MDB Section Format

The Music Data Base is the third - and last - Yamaha-chunk in a style file following the OTS data.

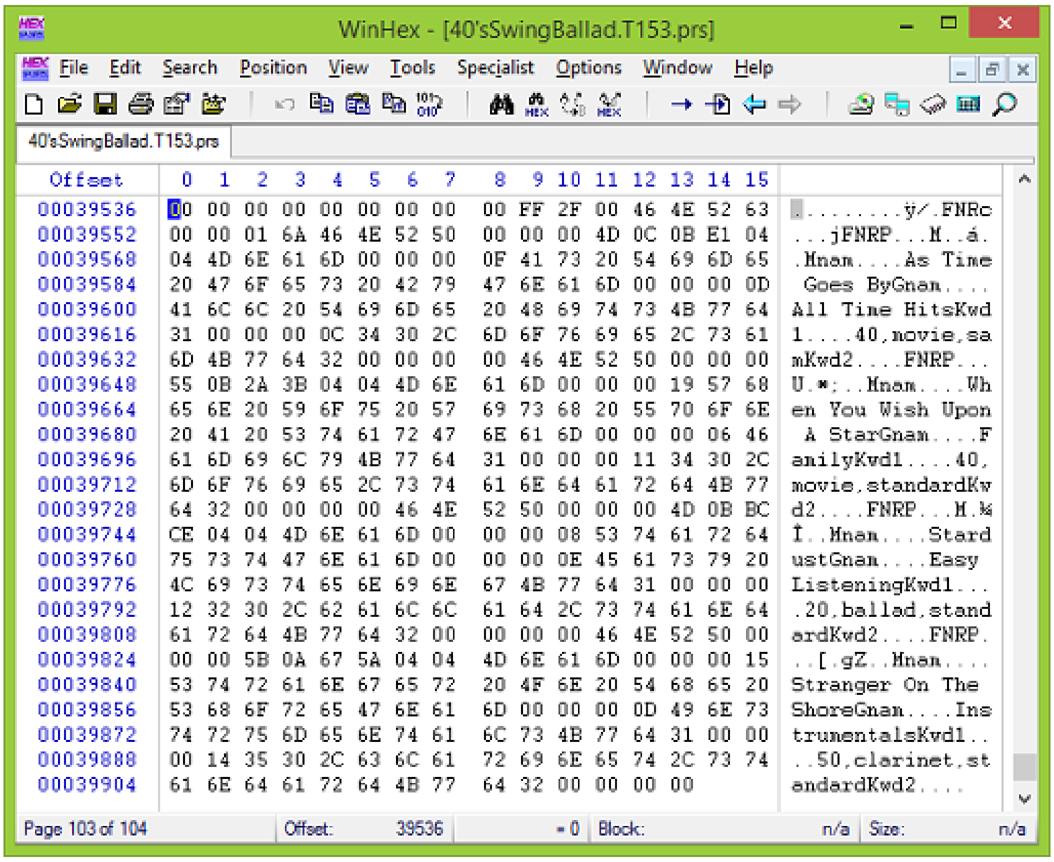
MDB Editor software program at http://www.jososoft.dk/yamaha/software/mdbedit/index.htm

This Music Data Base is stored in the style file, and must not be mixed up with Music Finder Data, which is stored in the keyboard. Read later in this document.

The MDB section has a chunk identifier; and a number of records.

To demonstrate, this is the MDB section in a 4 style file; which opened in a hex editor looks like this:

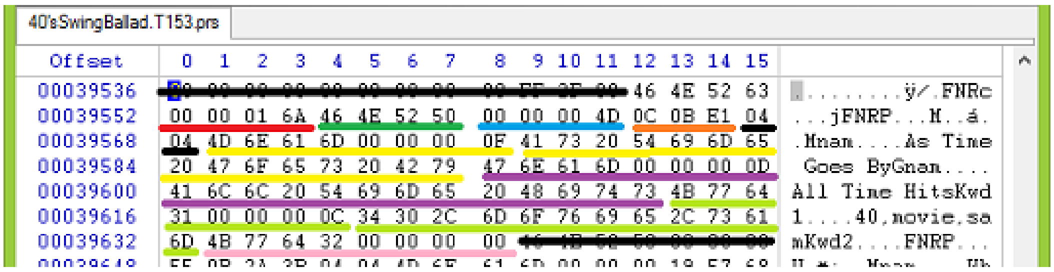
TIP:
Read Hex Numbers at http://www.jososoft.dk/yamaha/articles/midi_11.htm for converting between hexadecimal and decimal numbers.


WinHex - [40'sSwingBallad.T153.prs]
File Edit Search Position View Tools Specialist Options Window Help
40'sSwingBallad.T153.prs

```
  Offset     0  1  2  3  4  5  6  7   8  9 10 11 12 13 14 15
00039536    00 00 00 00 00 00 00 00  00 FF 2F 00 46 4E 52 63   ........ÿ/.FNRc
00039552    00 00 01 6A 46 4E 52 50  00 00 00 4D 0C 0B E1 04   ...jFNRP...M..á.
00039568    04 4D 6E 61 6D 00 00 00  0F 41 73 20 54 69 6D 65   .Mnam....As Time
00039584    20 47 6F 65 73 20 42 79  47 6E 61 6D 00 00 00 0D    Goes ByGnam....
00039600    41 6C 6C 20 54 69 6D 65  20 48 69 74 73 4B 77 64   All Time HitsKwd
00039616    31 00 00 00 0C 34 30 2C  6D 6F 76 69 65 2C 73 61   1....40,movie,sa
00039632    6D 4B 77 64 32 00 00 00  00 46 4E 52 50 00 00 00   mKwd2....FNRP...
00039648    55 0B 2A 3B 04 04 4D 6E  61 6D 00 00 00 19 57 68   U.*;..Mnam....Wh
00039664    65 6E 20 59 6F 75 20 57  69 73 68 20 55 70 6F 6E   en You Wish Upon
00039680    20 41 20 53 74 61 72 47  6E 61 6D 00 00 00 06 46    A StarGnam....F
00039696    61 6D 69 6C 79 4B 77 64  31 00 00 00 11 34 30 2C   amilyKwd1....40,
00039712    6D 6F 76 69 65 2C 73 74  61 6E 64 61 72 64 4B 77   movie,standardKw
00039728    64 32 00 00 00 00 46 4E  52 50 00 00 00 4D 0B BC   d2....FNRP...M.¼
00039744    CE 04 04 4D 6E 61 6D 00  00 00 08 53 74 61 72 64   Î..Mnam....Stard
00039760    75 73 74 47 6E 61 6D 00  00 00 0E 45 61 73 79 20   ustGnam....Easy 
00039776    4C 69 73 74 65 6E 69 6E  67 4B 77 64 31 00 00 00   ListeningKwd1...
00039792    12 32 30 2C 62 61 6C 6C  61 64 2C 73 74 61 6E 64   .20,ballad,stand
00039808    61 72 64 4B 77 64 32 00  00 00 00 46 4E 52 50 00   ardKwd2....FNRP.
00039824    00 00 5B 0A 67 5A 04 04  4D 6E 61 6D 00 00 00 15   ..[.gZ..Mnam....
00039840    53 74 72 61 6E 67 65 72  20 4F 6E 20 54 68 65 20   Stranger On The 
00039856    53 68 6F 72 65 47 6E 61  6D 00 00 00 0D 49 6E 73   ShoreGnam....Ins
00039872    74 72 75 6D 65 6E 74 61  6C 73 4B 77 64 31 00 00   trumentalsKwd1..
00039888    00 14 35 30 2C 63 6C 61  72 69 6E 65 74 2C 73 74   ..50,clarinet,st
00039904    61 6E 64 61 72 64 4B 77  64 32 00 00 00 00         andardKwd2....
```

Page 103 of 104 | Offset: 39536 | = 0 | Block: n/a | Size: n/a

Lets take a closer look at the section!



```
  Offset     0  1  2  3  4  5  6  7   8  9 10 11 12 13 14 15
00039536    00 00 00 00 00 00 00 00  00 FF 2F 00 46 4E 52 63   ........ÿ/.FNRc
00039552    00 00 01 6A 46 4E 52 50  00 00 00 4D 0C 0B E1 04   ...jFNRP...M..á.
00039568    04 4D 6E 61 6D 00 00 00  0F 41 73 20 54 69 6D 65   .Mnam....As Time
00039584    20 47 6F 65 73 20 42 79  47 6E 61 6D 00 00 00 0D    Goes ByGnam....
00039600    41 6C 6C 20 54 69 6D 65  20 48 69 74 73 4B 77 64   All Time HitsKwd
00039616    31 00 00 00 0C 34 30 2C  6D 6F 76 69 65 2C 73 61   1....40,movie,sa
00039632    6D 4B 77 64 32 00 00 00  00 46 4E 52 50 00 00 00   mKwd2....FNRP...
```

At position 12 in the first line comes the 4 byte chunk identifier = "FNRc"

Then the 4 byte length (red underlined) of the entire MDB chunk - not including the chunk identifier and this 4 "length" bytes.
In this case: "00 00 01 6A" (hex) = (0 * 256 * 256 * 256) + (0 * 256 * 256) + (1 * 256) + 106 = **362** bytes.

Next is the record identifier "FNRP" (green underlined) and the length of the record (blue underlined) - not including the record identifier and this 4 "length" bytes.
The record length is 4D (hex) = **77** (dec) bytes.

Then 3 bytes "0C 0B E1" (orange underlined) which defines the tempo calculated as:
Tempo = 60000000 / (1.byte * 256 * 256 + 2.byte * 256 + 3.byte)
In this case - when remembering to convert the hexadecimal values to decimal - the result is:
Tempo = 60000000 / (12 * 256 * 256 + 11 * 256 + 225) = 60000000 / 789473 = **76**

Next 2 bytes "04 04" (black underlined) which defines the Time Signature. In this case **4/4** - no calculations required...!

Next 8 bytes "4D 6E 61 6D 00 00 00 0F" (yellow underlined) which is the Music Name (Song Title) identifier "Mnam" and the length of Music Name. In this case "0F" (hex) = **15** (dec)

Next the 15 bytes defining the song title = **"As Time Goes By"**.

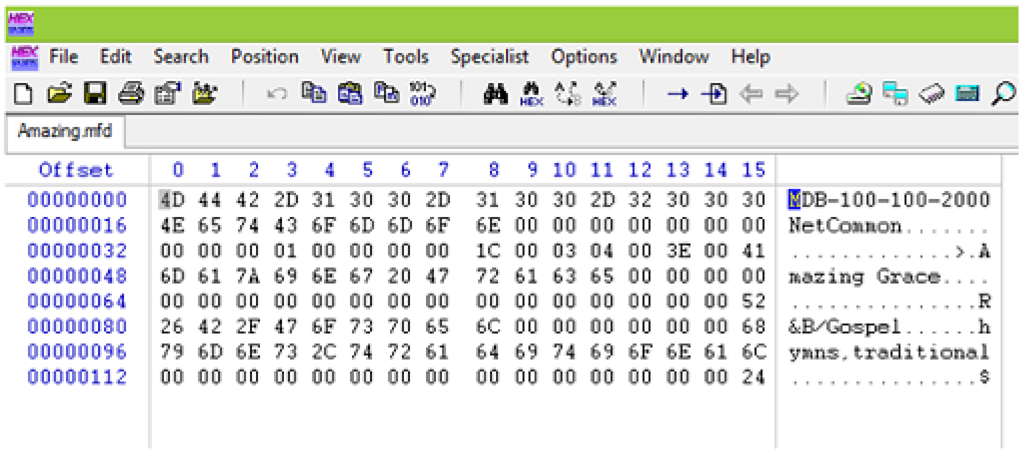
In a similar way comes
- Genre (identifier "Gnam") - violet underlined. The record reads: "All Time Hits"
- Keyword1 (identifier "Kwd1") - light green underlined. The record reads: "40,movie,sam"
- Keyword2 (identifier "Kwd2") - pink underlined. The record reads: ""

Notice that there is no data in Keyword2 (data length = 0).

1.5 Music Finder File Format

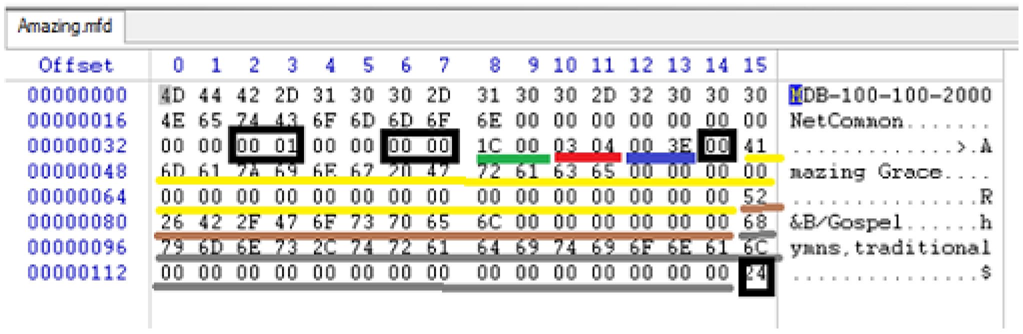
A Music Finder File (file extension: mfd) has a header; and a number of records.

To demonstrate, this test file is a single record Music Finder file; which opened in a hex editor looks like this:



```
Offset     0  1  2  3  4  5  6  7   8  9 10 11 12 13 14 15
00000000  4D 44 42 2D 31 30 30 2D  31 30 30 2D 32 30 30 30  MDB-100-100-2000
00000016  4E 65 74 43 6F 6D 6D 6F  6E 00 00 00 00 00 00 00  NetCommon.......
00000032  00 00 00 01 00 00 00 00  1C 00 03 04 00 3E 00 41  ..............>.A
00000048  6D 61 7A 69 6E 67 20 47  72 61 63 65 00 00 00 00  mazing Grace....
00000064  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 52  ...............R
00000080  26 42 2F 47 6F 73 70 65  6C 00 00 00 00 00 00 68  &B/Gospel......h
00000096  79 6D 6E 73 2C 74 72 61  64 69 74 69 6F 6E 61 6C  ymns,traditional
00000112  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 24  ...............$
```

This file has only ONE record. If the file was a multi record file, the following records are added at the end of this file.



```
Offset     0  1  2  3  4  5  6  7   8  9 10 11 12 13 14 15
00000000  4D 44 42 2D 31 30 30 2D  31 30 30 2D 32 30 30 30  MDB-100-100-2000
00000016  4E 65 74 43 6F 6D 6D 6F  6E 00 00 00 00 00 00 00  NetCommon.......
00000032  00 00 00 01 00 00 00 00  1C 00 03 04 00 3E 00 41  ..............>.A
00000048  6D 61 7A 69 6E 67 20 47  72 61 63 65 00 00 00 00  mazing Grace....
00000064  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 52  ...............R
00000080  26 42 2F 47 6F 73 70 65  6C 00 00 00 00 00 00 68  &B/Gospel......h
00000096  79 6D 6E 73 2C 74 72 61  64 69 74 69 6F 6E 61 6C  ymns,traditional
00000112  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 24  ...............$
```

The header is the first 36 bytes. Yamaha has used several formats, some keyboard specific and some of the "NetCommon" type.

As the format has never been published the idea about this has never been fully revealed. However most keyboards read Music Finder files with all headers.

The last two bytes in the header (index 34 and 35 in the first black square) are the total number of records in the file. In this case this number = 1, which is written 00 01.

Now comes the record(s). Each record is 92 bytes long.

# of bytes	Function	In this file	Marking
2	Two blanks	00 00	None
2	Record serial number (0-indexed) *)	00 00	Black square
2	Internal style number **)	1C 00 (= 7168 in dec.)	Green underline
2	Time signature	03 04 (= 3/4 time)	Red underline
2	Tempo	00 3E (= 62 in dec.)	Blue underline
1	Fav, S1 and S2 ***)	00	Black square
32	Song title	41 6D 61... (Ama...)	Yellow underline
16	Genre	52 26 42... (R&B...)	Brown underline
32	Keywords	68 79 6D... (hym...)	Gray underline
1	Intro/Next setting ****)	24 (= 36 in dec.)	Black square

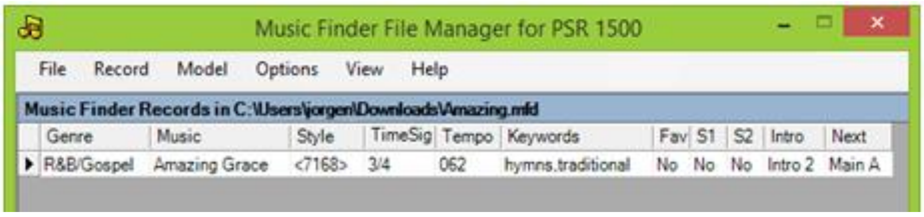
*) if there were a record more in the file, the serial number of this record would read 00 01
**) to get "Internal Style Number" please read here. Sorry there is no easier way!
***) "Fav" = Favorites (value = 1); "S1" = Search 1 (value = 2); and "S2" = Search 2 (value = 4). Values are added. E.g. value = 5 means: Favorites = Yes and Search 2 = Yes

****) "Intro" = Which intro will be played at start; "Next" = Which part will follow. Values are added.

Intro values	Next values
Off = 0 * 16	Off = 0
Intro 1 = 1 * 16	Intro 1 = 1
Intro 2 = 2 * 16	Intro 2 = 2
Intro 3 = 3 * 16	Intro 3 = 3
Main A = 4 * 16	Main A = 4
Main B = 5 * 16	Main B = 5
Main C = 6 * 16	Main C = 6
Main D = 7 * 16	Main D = 7
Ending A = 8 * 16	Ending A = 8
Ending B = 9 * 16	Ending B = 9
Ending C = 10 * 16	Ending C = 10

E.g. value = 24 (hex) = 36 (dec) = 2 * 16 + 4 means: Play "Intro 2" and continue with "Main A".

In the Music Finder File Manager software at http://www.josoSoft.dk/yamaha/software/mffm/index.htm the sample file will look like this:


Music Finder File Manager for PSR 1500

File Record Model Options View Help

Music Finder Records in C:\Users\jorgen\Downloads\Amazing.mfd

Genre	Music	Style	TimeSig	Tempo	Keywords	Fav	S1	S2	Intro	Next
R&B/Gospel	Amazing Grace	<7168>	3/4	062	hymns,traditional	No	No	No	Intro 2	Main A

Notice that the Internal Style Number is shown as <7168>. This means that the selected keyboard model (PSR 1500 - see top line in image above) does not recognize the style defined in the Music Finder record.

1.6 Multi Pad Format

A Multi Pad file is a format 1 MIDI file containing **5 tracks**.
The Multi Pad file has resolution of 96 ticks at smaller/older keyboards (e.g. PSR 740); but Tyros Multi Pads has a resolution of 1920.
The format differs slightly between models.

1.6.1 Tyros

The first track (Track 0) contains 10 MIDI Text events ALL placed at MIDI tick 0.
• 1. text event has a length of 6:
"CMxxxx", where xxxx corresponds to Chord Match value in Multi Pad 1, 2, 3, and 4.
Values for x are: 0 = No and 1 = Yes
• 2. text event has a length of 6:
"RPxxxx", where xxxx corresponds to Repeat value in Multi Pad 1, 2, 3, and 4.
Values for x are: 0 = No and 1 = Yes
• 3. - 6. text event has a length of 52:
This holds the Multi Pad name and number, e.g. "NyHipHop1 yxxxx...",
where y is the Multi Pad number, e.g. "N2HipHop1 2 ....". Rests of the 52 bytes are blanks.
• 7. - 10. text event has a length of 6:
This holds information about the image attached to the MultiPad.
E.g. "IyS375": "I" indicates image; "y" is the Multi Pad number; and "S375" is a reference to the image.

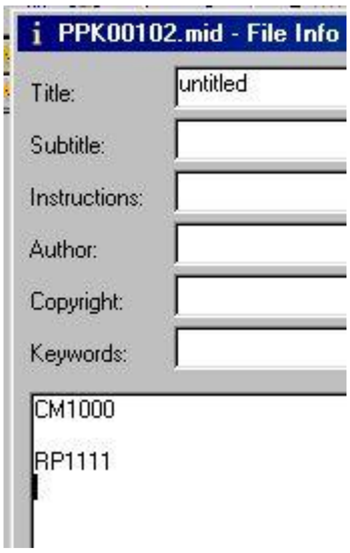
The following tracks (Track 1 - 4) are normal MIDI tracks with a length of one measure.
• The tracks contain voice definitions and note events.
• In Track 1 channel 1 is used; in Track 2 channel 2 is used; and so on.

1.6.2 Other models

Save a Multi Pad from your keyboard to a file. Open this file in your sequencer software and study the structure.

In Sonar (CakeWalk) you set the resolution in the Project Menu.

There is no "Loop" option when you record your PADs on the keyboard, but with Sonar (CakeWalk) you can set "Loop" command for PADs:


PPK00102.mid - File Info

Title: untitled

Subtitle:

Instructions:

Author:

Copyright:

Keywords:

CM1000

RP1111

- CM - means "Chord Match"
- RP - means "Repeat"

The numbers after them are "Flags":

- 0 - means "NO" for each of 4 PAD-buttons
- 1 - means "YES" for each of 4 PAD-buttons

1.7 Voice File Format

New voices can be obtained by modifying the voice samples in the keyboard - and for newer/bigger models by loading new voice samples.

Modifying samples are done by creating a voice file (file extensions .vce, .liv, .swv, .clv, .cvn, .org, .sar etc).

New samples are wav files, which have been loaded into the instrument (file extensions .TVN or .UVN for the last instruments). This kind of voices are not further discussed in this article.

1.7.1 Selecting a voice sample

The voice files are edits of existing preset voices.

Select them from the Voice Selection screen as you would any other voice, after navigating to the device (e.g. USB) that they are stored on.

1.7.2 Creating a voice sample

To create a Voice file do the following:

- Select a Voice for R1 or R2,
- You will see "Voice Set" at the bottom of the screen (on the S910 this is button 6)
- Adjust whatever sound parameters you wish
- Select "Save" (select the User drive or USB drive)
- Give your custom Voice a new name
- Complete the save operation

1.7.3 What is inside a voice sample

The voice file is actually a Standard MIDI File that stores information only, no sound/audio. Take a look at the Event List to the right.

When editing, you adjust parameters. Take a look in your keyboard manual. There you will find these parameters listed.

1.7.4 Computer programs for modifying voices

Edits can be done in the keyboard. But voice files can be created from any MIDI or style file; and modified in software too.

Take a look at MIDI Player II at http://psrtutorial.com/MB/midiplayer.html and MixMaster. at http://www.psrtutorial.com/MB/mixMaster.html Both are excellent programs written by Michael Bedesem.

Notice:
For audition of voice modifications always use your keyboard. Hardly any computer sound card are as good as your keyboard's sound card.

1.8 Yamaha File Extensions

This article describes the files extensions used by Yamaha keyboards.

1.8.1 Style Files - sty, pst, pcs, sst, prs, bcs, fps, scp, aus

Styles are the main component for music reproduction in arranger keyboards.

Style files holds MIDI data and some non-MIDI data. **NB: "aus"** styles are audio styles.

Style files had some years ago always **sty** file extension. Now a number of files extensions are used: **pst** (pianist); **pcs** (piano combo); **sst** (session); **prs** (pro); **bcs** (basic); **fps** (free play) and **scp** (dj style).

Beside the built-in styles additional style files can be loaded into most models. But due to some changes in the style file format during the years, styles might need conversion for use in smaller/older models.

1.8.2 Registration Files - rgt, reg

Registrations are saved snapshots of all active keyboard settings; e.g. style, instruments. This makes it possible to set up your keyboard pressing one button.

Registrations are saved to files with **rgt** or **reg** file extension. "reg" is an old format.

For the Tyros models Yamaha has developed utilities for converting registrations from one model to another.

More software tools for manipulation and conversion of registrations are available.

1.8.3 Voice Files - org, vce, liv, swv, clv, mgv, sar, sa2, ldr, drm, swv, nlv, mgv, sfx, lsf, env, cvn/d, cwn/d, uvn/d/i, tvn/d/i, vv1, cv1, vli, t2e

There are two different ways of changing the voices: Modifying the built-in voices or loading new voice samples.

The built-in voices can be modified, and may also work on other models. The new voice settings are saved to a file. File extensions are **org**; **vce**; **liv**; **swv**; **clv**; **mgv**; **sar**; **ldr**; and **drm**. Data format is MIDI.

When a built-in voice in the keyboard is loaded, the new settings are read and the voice is changed accordingly.

The latest high end arrangers, the Genos and Tyros models, will load 'real' new voices in the keyboard firmware. These voices are in files with **cvn/cvd**; **uvn/uvd** or **tvn/tvd** file extensions ("?vd" for drum kits); and these are not interchangeable between models.

1.8.4 Yamaha Expansion Packs (YEPs) - yep, ppi

A Yamaha Expansion Pack is a package which adds more content (voices and styles) to the keyboard.

The voices and styles are merged into a file, having a **yep** or **ppi** extension.

Some YEPs contain Registrations and/or MultiPads too. These data are not included in the "yep/ppi" file.

1.8.5 Music Finder Files - mfd

The Music Finder gives an easy way to select the style; tempo; and intro style part for songs.

The Music Finder holds a database of song records, where this information is stored.

Each model has a preset Music Finder database. The records in this database can be modified or deleted; and new ones can be added.

Furthermore the database can be replaced with a new database file. The file extension is **mfd**.

Music Finder database are keyboard specific; but to some extent they can be used in 'close' models. If not, software programs can be used edit the database on your PC.

1.8.6 MIDI and Audio Files - mid, wav, mp3

All Yamaha keyboards will read MIDI files - file extension **mid**. MIDI files holds data - 'the music recipe' - to the sound generator; but no 'music'.

This must not be confused with audio files - file extensions **wav** and **mp3**. This later holds 'music' and is read only by high end arrangers.

1.8.7 Multipad Files - pad, pd2

As an extra spice to your performances one of the built-in multipads can be added.

Besides this you can load additional multipads and get even more and new "spices".

Multipad files have a **pad** or **pd2** file extension. "pad" is a file in MIDI format, while "pd2" is an audio file link.

"pad" files can be created and edited in MIDI sequencer software.

Multipad files can be transferred from one keyboard to another; and you can replace the multipad in a style with another and save this.

1.8.8 Misc. files - usr, bup

usr = User data files
bup = Backup files (PSR E-models)
prg = Firmware Update File
msu = MIDI Setup File (System Reset Display)
ssu = System Setup File (System Reset Display)

1.8.9 Outdated files - ots, vic, eff

Files with an **ots** extension are 'One Touch Setting' files for PSR 9000 only. Later models use OTS data integrated in the style file.

Files with a **vic** extension are PSR 9000 voice data files.

Files with an **eff** extension are PSR 9000 effect data files.

1.9 Yamaha Internal Data

When developing software for Yamaha keyboards (or anything else...) you will need information about internal data; and the format of these.

Yamaha uses a lot of internal data formats; and only documents and publishes very few. Mainly only the MIDI related data.

This means that I have to spend many hours of revealing the internal data; and where and how this information is stored in files. In front of a particular keyboard.

Luckily a number of volunteers - having a diversity of keyboards at their disposal - have helped me in this tedious job. But still some data remains to be revealed.

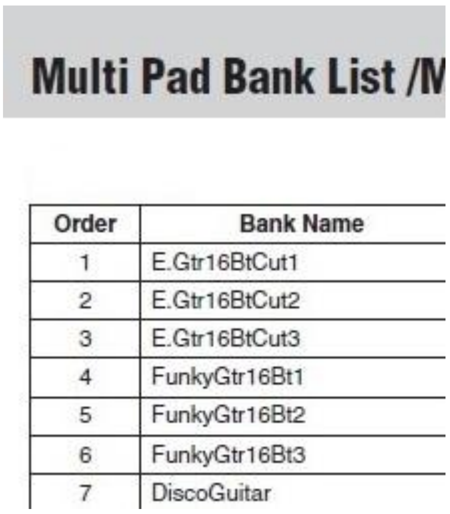
This page describes how you can assist me in updating my software programs for use in YOUR keyboard model.

I need Multi Pad numbers; Harmony/Arpeggio numbers; and Style numbers for various models.

1.9.1 Multi Pad Numbers

For use in OTS Editor at http://www.jososoft.dk/yamaha/software/otseditor/index.htm and OTS Viewer at http://www.jososoft.dk/yamaha/software/otsviewer/index.htm software programs I miss the **internal** Multi Pad Numbers for Genos; PSR A3000; PSR A2000; PSR OR700 and PSR S670 keyboards. (Note: The internal Multi Pad Numbers are **not** the numbers in the manual.)

To get the internal numbers, go to the MultiPad data page in your Manual. You will see something like:



Multi Pad Bank List /M

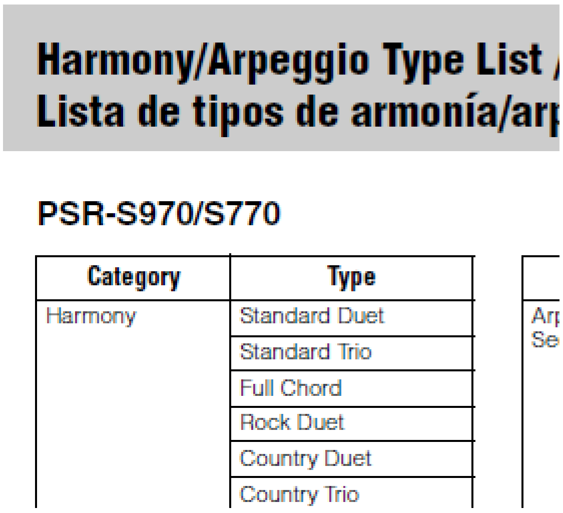
Order	Bank Name
1	E.Gtr16BtCut1
2	E.Gtr16BtCut2
3	E.Gtr16BtCut3
4	FunkyGtr16Bt1
5	FunkyGtr16Bt2
6	FunkyGtr16Bt3
7	DiscoGuitar

1. Now open a style (any will do)
2. Select One Touch Setting 1
3. Change the MultiPad to MultiPad number **1** in the manual data sheet (e.g. E.Gtr16BtCut1)
4. Select One Touch Setting 2
5. Change the MultiPad to MultiPad number **2** in the manual data sheet (e.g. E.Gtr16BtCut2)
6. Select One Touch Setting 3
7. Change the MultiPad to MultiPad number **3** in the manual data sheet (e.g. E.Gtr16BtCut3)
8. Select One Touch Setting 4
9. Change the MultiPad to MultiPad number **4** in the manual data sheet (e.g. FunkyGtr16Bt1)
10. Save the style as a user style as **mp1-4.sty**
11. Now repeat this sequence; but Change the MultiPad to MultiPad no **5**, **6**, **7** and **8**
12. Save the style as a user style as **mp5-8.sty**
13. Now repeat this sequence; but Change the MultiPad to MultiPad no **9**, **10**, **11** and **12**
14. Save the style as a user style as **mp9-12.sty**
15. etc.
16. Mail all the saved user styles to me. Find my address in the top right corner of this page.
17. I will now be able to extract the internal MultiPad numbers for all MultiPads.

1.9.2 Harmony/Arpeggio Numbers

For use in OTS Editor at http://www.jososoft.dk/yamaha/software/otseditor/index.htm software program I miss the **internal** Harmony/Arpeggio Numbers for Genos; PSR A3000; CVP 709/705/701. (Note: The internal numbers are **not** the order numbers in the manual.)

To get the internal numbers, go to the Harmony/Arpeggio data page in your Manual. You will see something like:



Harmony/Arpeggio Type List /
Lista de tipos de armonía/ar

PSR-S970/S770

Category	Type
Harmony	Standard Duet
	Standard Trio
	Full Chord
	Rock Duet
	Country Duet
	Country Trio

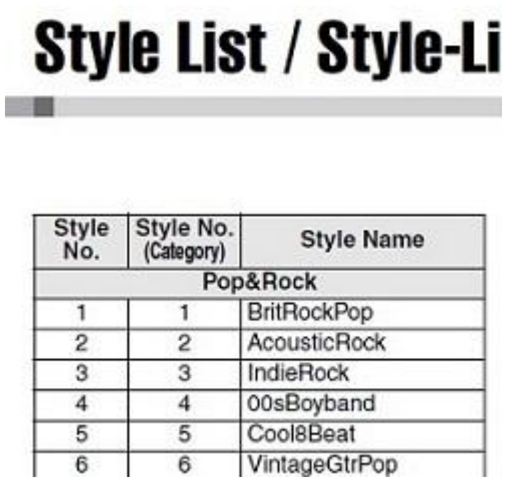
Arp
Se

1. Now open a style (any will do)
2. Select One Touch Setting 1
3. Change the Harmony to Harmony number **1** in the manual data sheet (e.g. Standard Duet)
4. Select One Touch Setting 2
5. Change the Harmony to Harmony number **2** in the manual data sheet (e.g. Standard Trio)
6. Select One Touch Setting 3
7. Change the Harmony to Harmony number **3** in the manual data sheet (e.g. Full Chord)
8. Select One Touch Setting 4
9. Change the Harmony to Harmony number **4** in the manual data sheet (e.g. Rock Duet)
10. Save the style as a user style as **ha1-4.sty**
11. Now repeat this sequence; but Change the Harmony to Harmony no **5**, **6**, **7** and **8**
12. Save the style as a user style as **ha5-8.sty**
13. Now repeat this sequence; but Change the Harmony to Harmony no **9**, **10**, **11** and **12**
14. Save the style as a user style as **ha9-12.sty**
15. etc.
16. Mail all the saved user styles to me. Find my address in the top right corner of this page.
17. I will now be able to extract the internal Harmony/Arpeggio numbers for all Harmony/Arpeggio.

1.9.3 Style Numbers

For use in Music Finder File Manager at http://www.jososoft.dk/yamaha/software/mffm/index.htm software program I miss the **internal** Style Numbers for CVP 709; CVP 705; CVP 601; PSR A3000; PSR A2000; PSR OR700; PSR S650; PSR S550; and PSR 550 keyboards. (Note: The internal Style Numbers are **not** the numbers in the manual.)

To get the internal numbers, go to the Style data page in your Manual. You will see something like:



Style List / Style-Li

Style No.	Style No. (Category)	Style Name
Pop&Rock		
1	1	BritRockPop
2	2	AcousticRock
3	3	IndieRock
4	4	00sBoyband
5	5	Cool8Beat
6	6	VintageGtrPop

1. Now create a new Music Finder file
2. Add an entry in the Music Finder using the **first** style in the manual style list (e.g. BritRockPop)
3. Name the entry (Song Title) the same as the style (e.g. BritRockPop)
4. Add a new entry in the Music Finder using the **second** style in the manual style list (e.g. AcousticRock)
5. Name the entry (Song Title) the same as the style (e.g. AcousticRock)
6. Add another new entry in the Music Finder using the **third** style in the manual style list (e.g. IndieRock)
7. Name the entry (Song Title) the same as the style (e.g. IndieRock)
8. etc.
9. Mail the Music Finder file to me. Find my address in the top right corner of this page.
10. I will now be able to extract the internal Style numbers for all styles.

1.10 HP_MIDIFILE.dll in C# and VB.Net

The HP_MIDIFILE.dll file file at http://www.heikoplate.de/hpm/index.php?option=com_content&task=view&id=530&Itemid=68 from Heiko Plate - originally written for MIDI manipulation in C++ coded software programs - can be utilized in C# and VB.Net software programs as well. Two code samples are shown below.

For more information about the functions etc. in HP_MIDIFILE.dll please read the documentation from Heiko Plate.

The method is explained with a sample program that converts a MIDI type 1 file (c:\temp\test1.mid) to a MIDI type 0 file (c:\temp\new1.mid).

The sample programs consists of one form that has one command button called button1. The file HP_midifile.dll is located in the same folder as the programs.

The method used here is to declare the names of dll functions that will be needed in the programs using the internal identifications of the library. These "decorated names" are listed in the map-file HP_midifile.map.

The functions often refer to arguments such as HP_SMF0. The values expected by the dll are defined in the header file HP_midifile.h. For example the line

```
#define HP_SMF0    0    /* Standard MIDI file-format 0 */
```

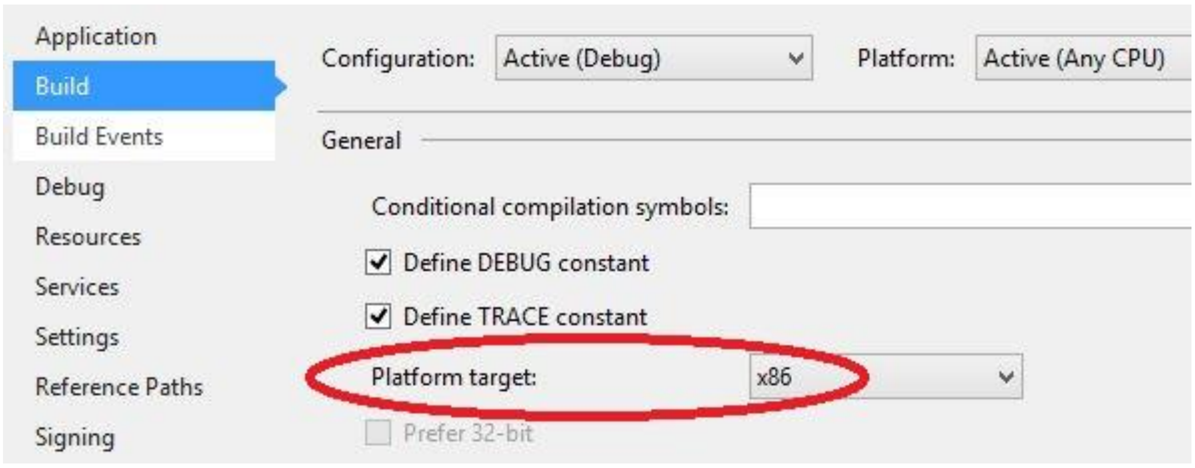
means passing a "0" signifies a type 0 MIDI file.

1.10.1 C# sample code for Form1

```
// all code for imports, UI etc.
// generated by the designer is omitted

using System.Runtime.InteropServices;

namespace HP_test
{
 public class Form1 : System.Windows.Forms.Form
 {

   [DllImport("HP_midifile.dll",
     EntryPoint="?HP_Init@@YAPAVMIDIFile@@XZ")]
     public static extern int HP_Init();

   [DllImport("HP_midifile.dll",
     EntryPoint="?HP_Load@@YAIPAVMIDIFile@@PBD@Z")]
     public static extern int HP_Load(int i, String s);

   [DllImport("HP_midifile.dll",
     EntryPoint="?HP_Save@@YAIPAVMIDIFile@@PBDH@Z")]
     public static extern int HP_Save(int i, String s, int x);

   [DllImport("HP_midifile.dll",
     EntryPoint="?HP_Free@@YAIPAVMIDIFile@@@Z")]
     public static extern int HP_Free(int i);

   private void button1_Click(object sender, System.EventArgs e)
   {
    int smf0 = 0;
    int hp_err_none = 0;
    int mf = HP_Init();
    if (mf == 0)
    {
             // do some error handling here
    }
    int result = HP_Load(mf, "c:\\temp\\test1.mid");
    if (result != hp_err_none)
    {
             // do some error handling here
```

```
      }
      result = HP_Save(mf, "c:\\temp\\new1.mid", smf0);
      if (result != hp_err_none)
      {
               // do some error handling here
      }
      result = HP_Free(mf);
      if (result != hp_err_none)
      {
               // do some error handling here
      }
    }
  }
}
```

Due to a 64 bit vs. 32 bit conflict open "Projects" -> "Properties" window in Visual Studio and set the Target CPU as x86. You will get some warnings while compiling!

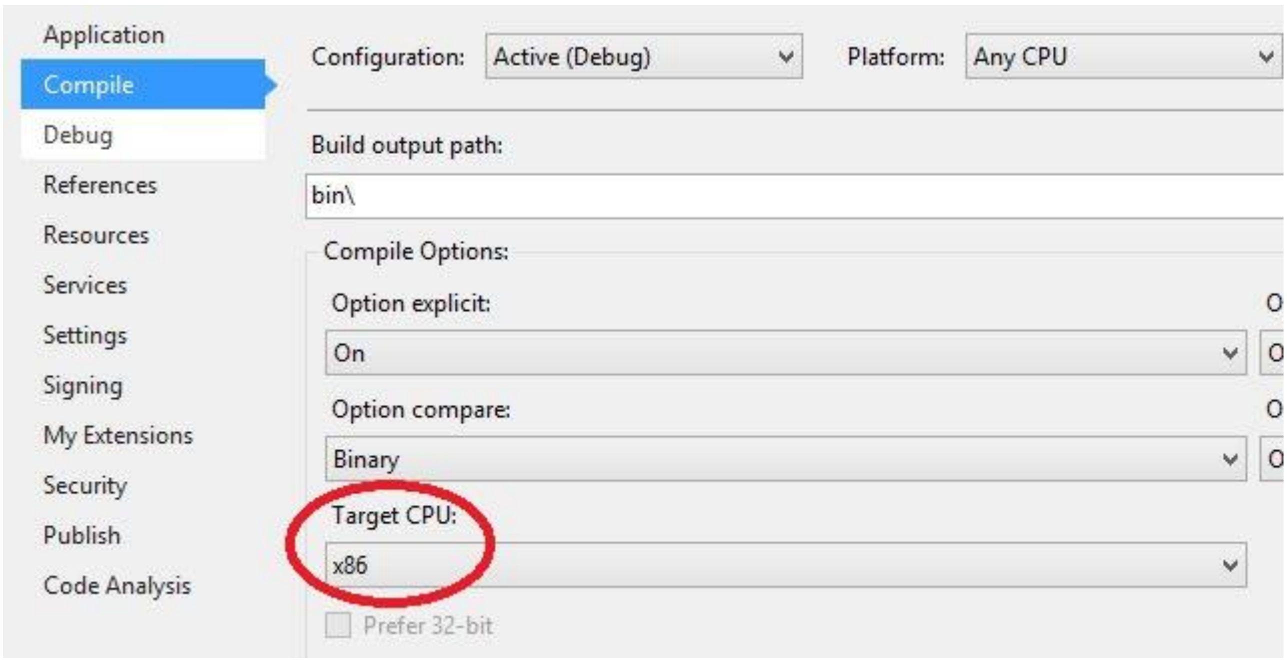


Configuration: Active (Debug)
Platform: Active (Any CPU)
Conditional compilation symbols:
☑ Define DEBUG constant
☑ Define TRACE constant
Platform target: x86
☐ Prefer 32-bit

1.10.2 VB.Net sample code for Form1

```
' all code for imports, UI etc.
' generated by the designer is omitted

Imports System.Runtime.InteropServices

Public Class Form1 Inherits System.Windows.Forms.Form

  Private Declare Function HP_Init Lib _
  "HP_midifile.dll" _
  Alias "?HP_Init@@YAPAVMIDIFile@@XZ" () As Integer

  Private Declare Function HP_Load Lib _
  "HP_midifile.dll" _
  Alias "?HP_Load@@YAIPAVMIDIFile@@PBD@Z" (ByVal i As _
  Integer, ByVal s As String) As Integer

  Private Declare Function HP_Save Lib _
  "HP_midifile.dll" _
  Alias "?HP_Save@@YAIPAVMIDIFile@@PBDH@Z" (ByVal i As _
  Integer, ByVal s As String, ByVal x As Integer) As Integer

  Private Declare Function HP_Free Lib _
  "HP_midifile.dll" _
  Alias "?HP_Free@@YAIPAVMIDIFile@@@Z" (ByVal i As Integer) _
  As Integer
  Private Sub Button1_Click(ByVal sender As System.Object, _
  ByVal e As System.EventArgs) Handles Button1.Click

    Dim smf0 As Integer = 0
    Dim HP_ERR_NONE As Integer = 0
    Dim mf As Integer = HP_Init()
    If mf = 0 Then
            ' do some error handling here
    End If
    Dim result As Integer = HP_Load(mf, "c:\temp\test1.mid")
    If Not result = HP_ERR_NONE Then
            ' do some error handling here
    End If
    result = HP_Save(mf, "c:\temp\new1.mid", smf0)
    If Not result = HP_ERR_NONE Then
            ' do some error handling here
    End If
    result = HP_Free(mf)
    If Not result = HP_ERR_NONE Then
            ' do some error handling here
    End If
  End Sub

End Class
```

Due to a 64 bit vs. 32 bit conflict open "Projects" -> "Properties" window in Visual Studio and set the Target CPU as x86. You will get some warnings while compiling!



Configuration: Active (Debug)
Platform: Any CPU
Build output path: bin\
Compile Options:
Option explicit: On
Option compare: Binary
Target CPU: x86
Prefer 32-bit