



XF Format Specifications

V 2.01

January 13, 1999

YAMAHA CORPORATION

Contents

§1 Outline of the format.....	3
§2 MIDI Specifications	6
1. Contents of the beginning of the track chunk.....	6
2. XF Information Header	9
3. Lyric Meta-Event	13
4. XF Karaoke Message.....	16
5. Style Messages	19
Appendix A. Character codes.....	26
Appendix B. Country codes	27
Appendix C. Song genre	29
Appendix D. Song keywords	30
Appendix E. XG Song Data Number	31

§1 Outline of the format

To meet the needs of a full-fledged multimedia environment, Yamaha has developed a new format for MIDI sequence data; the “XF format.” This new file format is an extension of the “Standard MIDI File” (SMF) format defined as MIDI RP-001 (Recommended Practice), which meets the continually increasing sophistication of the computer environment, takes greater advantage of MIDI’s compressibility and interactive character, and allows upward compatibility of data.

Yamaha will continue to develop and market electronic musical instruments and software that uses the “XF format,” and will endeavor to maintain data compatibility and scalability between Yamaha models.

<Background of development>

MIDI is being used in a wide variety of fields other than musical instruments, such as communications and computer games. A sequencer is required in order to edit or playback a song using MIDI. However sequencer file formats differed between manufacturers and models, and could not be used compatibly. In 1988, the American MMA (MIDI Manufacturers Association) proposed a “Standard MIDI File” (SMF) format in order to unify sequencer file formats, and in 1991 the MIDI association adopted this as an RP (Recommended Practice). This meant that files could be exchanged between SMF-compatible sequencers. The number of MIDI users grew dramatically, and the variety of SMF-compatible software is also increasing. However, since the principle objective of SMF is playback of MIDI data, it provides only a minimal definition, and does not define song lyric data or score data, etc. Thus, there is an increasing need for expansion to meet these requirements.

In the area of computer-based multimedia, sound and video are receiving greater attention. Similarly to the progress in image compression being made in the field of video, developments are taking place in multimedia sound and music handling as well. At present, there are two approaches to the handling of multimedia sound. One method is for the software to digitally store the sound data, and play it back continually. The second method is to have the software supply control data that makes the computer's tone generator produce sound. The first method offers realistic sound, but requires immense quantities of data and does not allow the performance or voices to be changed freely. The second method requires less data and allows voices and tempo etc. to be changed freely, and is therefore suitable for interactive multimedia such as repetitive game music or karaoke. MIDI-based applications are typical of this second type of approach. This is the background out of which arises the need for a compatible sequencer file format with expanded capabilities. To respond to these needs, Yamaha proposes the “XF format” — the next-generation sequencer file format.

<Basic concepts>

“XF” is a new format which includes the flexibility and compatibility that made SMF a successful format, and goes on to provide even greater expressive power and flexible scalability, as well as data continuity.

This “XF” format has the following features:

- An “XF Information Header” is defined for noting features or attributes of the song, for convenience in searching and classifying.
- The song lyrics data meta-event “Lyrics Meta Event” which had not been defined by SMF in detail has been clearly defined, allowing karaoke support.
- Compatibility with the “Standard MIDI Files with Lyrics” specification of the TUNE1000 Corp. (Canada).
- A special banner sequence data type, the “XF Karaoke Message” targeted at banner display for karaoke applications etc. has been added.
- In the future, it is expected that expansions will be made to allow sampled data and MIDI to be linked.

The “XF format” is based on the following three principles.

- Compatibility
- Scalability
- Expandability

1. Compatibility

“XF format” compatible song data can be used on any “XF format” compatible sequencer.

2. Scalability

Since the “XF format” is based on SMF, the basic aspects of MIDI such as performance data are supported by SMF, and extensions can be freely supported by sequencer or other processing device.

3. Expandability

The “XF format” will be flexibly extended in line with future product development.

<Extensions to SMF>

- An “XF Information Header” has been defined and added to list features/attributes of the song.
- The “Lyric Meta-Event” which was loosely defined by SMF has been defined in detail.
- The addition of “XF Karaoke Message,” a special banner sequence data type targeted at karaoke applications etc., and defined based on the Lyrics meta event and Cue Point meta-event.

<Data structure>

The data structure of the XF format is based on the data structure of SMF.

Events that are newly defined by the XF format use SMF meta-events.

This means that the XF format can accommodate lyric data and karaoke data while preserving the SMF data structure.

Several chunks are defined as additions. By using added chunks, the data structure is made functionally cleaner.

Examples of the added chunks are shown below.

XF format data file structure

MThd.....	Header Chunk (SMF)
MTrk.....	Track Chunk (SMF)
..(information header).....	
... (karaoke).....	
..... (karaoke).....	
..... (karaoke).....	
.....	
XFIH... (information header).	XF Information Header Chunk
.....	
XFKM... (karaoke).....	XF Karaoke Message Chunk
... (karaoke).....	
..... (karaoke).....	
.....	

The XF Information Header chunk is a cluster of data that begins with the symbol “XFIH.” Characteristics and attributes of the song are placed here.

The XF Karaoke Message chunk is a cluster of data that begins with the symbol “XFKM.” Lyrics and screen wipe data etc. are placed here.

It is also possible to divide the XF chunks into separate files.

If the XF Information Header chunk is divided into a separate file, it uses a filename extension of “.XIH”

If the XF Karaoke Message chunk is divided into a separate file, it uses a filename extension of “.XKM”

§2 MIDI Specifications

1. Contents of the beginning of the track chunk

- SMF format 0 is used.
- The following meta-events defined by SMF are used to supplement the XF Information Header.

1) Song Name (Sequence/Track Name)

Uses the SMF-defined meta-event Sequence/Track Name
Alphabetically displayed song name

2) Meter (Time Signature)

Uses the SMF-defined meta-event Time Signature
SMF-defined time signature

3) Tempo (Set Tempo)

Uses the SMF-defined meta-event Set Tempo
SMF-defined tempo

- In addition to these, the following data is unique to the XF format.

4) XF Version ID

Uses the SMF-defined meta-event Sequencer-Specific Meta-Event
ID that shows the XF format version

5) XF Lyrics Header

Uses the SMF-defined meta-event Cue Point
Data for subsequent lyric meta-events

<Song Name (Sequence/Track Name)>

- In current song data, it is usual to use the SMF meta-event — Sequence/Track Name to input the song name. The XF format uses the meta-event — Sequence/Track Name as the song name
- Since the character code attribute cannot be specified, this is written in ASCII code (alphabetical notation).
- This is specified at the beginning of the SMF track chunk, at delta-time=0 (the timing of 1 : 1 : 0).
- The processing system will be expected to search for this event at least as far as the first key-on event.

FF 03 len <text> Sequence/Track name
 len:data length

Example:

Song name "After Hours Club"

FF 03 10 41 66 74 65 72 20 48 6F 75 72 73 20 43 6C 75 62

<Meter (Time Signature)>

- The initial value must be specified at the beginning of the SMF track chunk, delta-time=0 (timing of 1 : 1 : 0).
- The processing system will be expected to search for this event at least until the first key-on event.

FF 58 04 nn dd cc bb Time Signature
 nn: numerator
 dd: denominator

Indicates the power of two to which the denominator corresponds. A value of 2 is quarter note, and a value of 3 is eighth note.

cc: number of MIDI clocks in a metronome click
 The number of MIDI timing clocks (24 tpqn) in one count of the metronome. In this example, the metronome would sound every quarter note.

bb: number of notated 32nd-notes in a MIDI quarter-note
 Indicates the number of notated 32nd notes that correspond to one MIDI quarter note.

Example:

4/4 time signature
 FF 58 04 04 02 18 08

<Tempo (Set Tempo)>

- The initial value must be specified at the beginning of the SMF track chunk, at delta-time=0 (a timing of 1 : 1 : 0).
- The processing system will be expected to search for this event at least as far as the first key-on event.

FF 51 03 tt tt tt Set Tempo
 tt tt tt: microseconds per MIDI quarter-note
 μ (micro) seconds / MIDI quarter note

Example:

120bpm
 FF 51 03 07 A1 20 60,000,000 μ seconds / 120 = 500,000
 (7 × 256 × 256) + ((10 × 16 + 1) × 256) + (2 × 16) = 500,000

<XF Version ID>

- To provide for compatibility in the event of future expansion, the XF format version ID and status must be listed.
- This is required by the XF format.
- This is specified in the format of a Sequencer-Specific meta-event.
- This is inserted at the beginning of the track chunk, at delta-time=0 (a timing of 1 : 1 : 0).
- The processing system will be expected to search for this event at least as far as the first key-on event.

FF 7F 09 43 7B 00 58 46 30 32 s1 s0

'XF02'(58 46 30 32) Version 2.0
 s1,s0 Status 00000000 000kl0si
 Each bit is a flag indicating the presence (1) or absence (0) of the corresponding data/chunk
 i: XF Information Header / XF Information Header Chunk
 l: Lyric Meta-Event
 k: XF Karaoke Message / XF Karaoke Message Chunk
 s: XF Style Message

<XF Lyrics Header>

- This indicates that the subsequent lyric meta-events (FF 05 len <text>) are lyric events compatible with the XF format,
 - the MIDI channel of the melody part,
 - the offset value for lyric display,
 - the character code system used to notate the lyrics,
 - etc.
- This is notated in ASCII. (alphabetical notation)
- This is specified in Cue Point format.
- This event is part of the XF karaoke message.
- This is placed at the beginning of the track chunk, at delta-time=0 (a timing of 1 : 1 : 0)
- The processing system will be expected to search for this event at least as far as the first key-on event.
- Data items are separated by a single-byte colon “:”

FF 07 len <text>

Data items (contents of <text>)

‘\$Lyrc’

XF Lyrics Header ID

An ID indicating that this is a XF Lyrics Header

Indicates that subsequent lyric meta-events are XF format compatible.

‘c,c,...’ c:MIDI Channel (decimal number as text)

MIDI Channel of Melody Part

More than one may be given. In this case, separate each by a comma “,”

Used to switch the melody part on/off etc.

‘o’ o:Offset Value (decimal number as text)

Lyrics Display Offset Value

When lyrics are displayed by paragraph, this indicates how far ahead of the performance timing the lyrics will be displayed.

This will be according to the resolution (time base) specified in the header chunk.

The processing system must keep processing of the lyric display wipe timing etc. ahead of the playback timing by the number of clocks specified here.

‘l’ l:Language

Language (character code data)

Data that specifies the character code system used for the lyrics.

Use the symbols given in “Appendix A. Character codes.”

If the character code listed in the Language field cannot be handled, lyric meta-events need not be displayed.

Example: XF lyric header when the MIDI channels of the melody are ch.4 and ch.12, the offset value is 240/480 tpqn, and the language of the lyrics is Japanese

FF 07 len <\$Lyrc:4,12:240:JP>

2. XF Information Header

- Information on the features/attributes of the song is specified in the form of a text meta-event. This is used for searching/classifying songs.
- The information header can be given in the following three ways, and handling is up to the processing system.

- 1) Specified as a meta-event within the SMF track chunk.
- 2) Specified in meta-event format, in a separate chunk following the track chunk.
 Chunk type: 'XFIH'
 Data length: 4 bytes
 <XFIH Event> = <delta-time> <Meta Event>
- 3) Specified with the chunk of 2) placed in a separate file.
 The contents of the file are identical to chunk 2).
 DOS filename extension: '.XIH'
 Mac file type: 'XFIH'

If two or more of these exist, they will take priority in the order of 3), 2), 1).

<XF Information Header Event>

- These provide basic attributes of the song. Items have been specified to make it easy to search for songs.
- This is placed immediately after the XF version ID at delta-time=0 (timing of 1.1.0).
- Specified in the form of a Text Event meta-event.
- The English header will be the common header, and a language specific header is also specified. One or more are used as necessary.
- The processing system must be able to handle cases in which only one or the other exists, or in which two or more exist.
- Each item is separated by a colon ":" and given in succession.
- Unlisted data items are left empty. (Colons will be entered in succession.)
- New categories are added following the last item. The processing system shall consider any following data items as blank when there is no more text, even if no colons are found.
- There are two types: XF Information Header — Common and XF Information Header — Language Specific.

FF 01 len <text>

<XF Information Header — Common>

- Given in ASCII (alphabetical notation).

Data items

- 1) XF Information Header —Common — ID
 An ID (4 characters) indicating that this is the XF Information Header — Common "XFhd"
- 2) Date (year/month/day of publication)
 The date that the original song was created or published.
 This is given in western reckoning, and year, month, and day are separated by a slash '/'.
 Example: 1994/09/28
 Example: 1994// (year only)
 Example: 1994/09/ (year and months only)
- 3) Country (country of production)
 The country in which the original song was created.
 Use the two-character symbol given in "Appendix B. Country codes."

- 4) **Category (genre of song)**
English text indicating the genre of the original song or the song data.
If two or more categories are given, separate them with a slash '/'.
Use the categories given in "Appendix C. Song categories."
- 5) **Beat**
English text indicating the beat of the rhythm of the original song or the song data.
Select from the following.
- 2 Beat, 2 Beat(March),
 - 3 Beat, Waltz, Waltz(Swing), Other 3/4,
 - 4 Beat, 4 Beat(Swing), Other 4/4,
 - 6 Beat, 12/8(6/8),
 - 8 Beat,
 - 12 Beat,
 - 16 Beat,
 - 24 Beat,
 - Shuffle(Bounce), Other(Even), Other(NonEven),
 - Others
- Since a clear indication of the beat is required in order to change the practice method for each song, the current Beat categories were retained, and the Other.. categories were added.
 - Use this when the current beat designation is insufficient for classification.
- 6) **Instrument on Melody part (main instrument for the melody part)**
The program change number of the main instrument that plays the melody part of the song data.
Select from the 128 voices of the GM Sound Set, and give in decimal form.
Use from 1 through 128. (1-128)
- 7) **Vocal Type**
This indicates the gender and number of the person(s) playing or singing the original song.
Select from the following.
- f1: Female Solo
 - m1: Male Solo
 - fm: Female & Male (male/female duet, or male/female mixed),
 - fp: Female Chorus,
 - mp: Male Chorus,
 - no: Instrumental (no vocal)
- 8) **Composer**
The name of the composer of the original song is given in alphabetical characters.
List in the order of Given Name and Family Name.
Separate the given and family names with a space.
A middle name may also be input.
If there are two or more names, separate them with a slash "/".
- Example: Jake Ryan
- 9) **Lyricist**
If the original song has lyrics, the name of the lyricist.
The format is the same as for Composer.

-
- 10) Arranger
The name of the person who arranged the original song or song data.
The format is the same as for Composer.
 - 11) Performer (performer or singer)
The name of the person or group who played or sang the original song.
The format is the same as for Composer.
 - 12) Programmer
The name of the person who produced the music data.
The format is the same as for Composer.
 - 13) Keyword
Input keywords chosen from Appendix D.
One or more may be input.
Use a slash “/” to separate multiple entries.

Example: movie/Love Song/Tears

<XF Information Header — Language Specific>

- Data items 1) and 2), and the various control characters are given in ASCII.
- Details of the XF Information Header—Language Specific for character sets other than ASCII are given below.

Data items

- 1) XF Information Header — Language Specific — ID
An ID (4 characters) indicating that this is the XF Information Header — Language Specific. “XFln”
- 2) Language
Data that specifies the character code system used in the XF information header (language specific).
This does not specify the character code system used for the lyrics. The character code system for the lyrics is specified by the XF lyric header. Nor does this specify the country in which the song data was produced. Use the symbols given in “Appendix A. Character codes.”
- 3) Song Name
The language-specific display of the song name.
If you wish to display the song name over two or more lines, insert a slash “/” at the desired line return locations.

Example: This is my song./This is your song.

Display : This is my song.
 This is your song.

If the processing system displays the song, it must interpret control characters as in the above example.

- 4) Composer
The name of the composer of the original song.
Separate the first name and last name with a single-byte space “ ”.
If there are two or more names, separate them with a single-byte slash “/”.

Example: Jake Ryan/Lydia Diaz

5) Lyricist

If the original song has lyrics, this is the name of the lyricist.

The format is the same as for the Composer.

6) Arranger

The name of the person who arranged the original song or the song data.

The format is the same as for the Composer.

7) Performer (performer/singer)

The name of the person or group performing or singing the original song.

The format is the same as for the Composer.

8) Programmer

The name of the person who produced the song data.

The format is the same as for the Composer.

Example: An American song published on September 28, 1994, where the genre is Pops, the beat is 8Beat, the melody played in the song data is by a sax, solo vocals are by a female, composed by Jake Ryan, lyrics by Kerry Williams, no arranger, performed and sung by Lydia Diaz, song data produced by Joe Moore and titled "Happy Sunday" would have the following header.

Example of XF Information Header (Common):

```
FF 01 len <XFhd:1994/09/28:JP:Pops:8Beat:65:fs:Jake Ryan:Lydia Diaz::  
Kerry Williams:Joe Moore:movie/Love Song>
```

Example of United States — XF Information Header (Language Specific):

```
FF 01 len <XFIn:US:Happy Sunday:Jake Ryan:Kerry Williams:Lydia Diaz:Joe  
Moore>
```

3. Lyric Meta-Event

- This defines the use of the Lyrics meta-event within the track chunk which had not been defined in detail by the SMF format.
- This is compatible with the Standard MIDI Files with Lyrics specification of the TUNE1000 Corp. (Canada).
- This is compatible with the “Proposal about Lyric Meta Events in Standard MIDI Files” by the Association of Musical Electronics Industry (AMEI).

<Syllable>

- The SMF-defined Lyric meta-event is used.
- Each syllable is a single Lyric meta-event.
- Add a single-byte space “ ” after the last syllable of a word.
- Add a single-byte space “ ” after the punctuation mark of the last syllable that has a punctuation mark.

FF 05 len <text>

Example: FF 05 len <This >
 FF 05 len <is >
 FF 05 len <an >
 FF 05 len <ex>
 FF 05 len <am>
 FF 05 len <ple. >

<Carriage Return Event>

- This is a special event which indicates the end of a line when the lyrics are displayed on screen etc.
- When this event occurs, a new line is begun in the lyrics display.

FF 05 01 0D

<Line Feed Event>

- This is a special event which indicates the end of a paragraph for the displayed lyrics.
- This event will clear the screen in order to display the following paragraph.

FF 05 01 0A

Control characters

- Lyric data may include control characters to perform various types of processing.
- ASCII (single-byte) characters are used for control characters.
- If you wish to display the control character itself, prefix it with a single-byte backslash “\”.

\r : Carriage Return(0D)
 \n : New Line (Line Feed)(0A)
 \t : Horizontal Tab (09)
 \\ : ‘\’ itself
 \{ : ‘{’ itself
 \} : ‘}’ itself
 \[: ‘[’ itself
 \] : ‘]’ itself

In some cases it is difficult to enter control codes (00-1F), so combinations of ‘\’ (backslash) and a following single-byte ASCII character are defined to indicate return (\r), new-line (\n), and horizontal tab (\t). Other combinations with a single-byte ASCII character are undefined.

Tags

- 1) Character code information tag (Suggested Character Information Tag)
This is used to suggest the character code. This must be given at the beginning of the Lyric meta-event. This data is valid until the next character code data.

Application: {@<keyword>}

{@LATIN},{@Latin},{@latin} G@default

Indicates the ANSI character code set for most European languages. If the character code is not specified, the lyrics will be handled as {@LATIN}.

If the character code given by the character code data cannot be handled by the processing system, display need not be performed until the arrival of character code data that can be handled.

Other keywords are reserved for other character sets.

If the FF FE or FE FF which indicates the UNICODE byte order exists at the beginning, the character code is handled as UNICODE.

- 2) Song information (Song Information Tag)
The Song Information Tag is used when you wish to display information relating to the song.
The keywords for each item are given enclosed between ‘#’ and ‘=’ .

Usage: @!# <item> = <content>

Each ‘!’ must be given as the beginning of each Lyric meta-event.

Item types:

@!#TITLE= or !#Title= or !#title=

Use this to display the song name or the name of the original song, etc.

Example: !#Title=Happy Sunday}

@!#COMPOSER= or !#Composer= or !#composer=

Use this to display the composer’s name.

Example: !#COMPOSER=Jake Ryan}

!#LYRICS= or !#Lyrics= or !#lyrics=

Use this to display the lyricist’s name.

Example: !#LYRICS=Lydia Diaz}

!#ARTIST= or !#Artist= or !#artist=

Use this to display the name of the singer or performer.

Example: !#Artist=Kerry Williams}

3) Null Tag

Usage: @{#}

Use this to explicitly indicate the tag boundary or completion.

Example: {#Title=Happy Sunday}
{#Composer=Jake Ryan}
{#Lyrics=Lydia Diaz}
{#Artist=Kerry Williams}
{#}

4. XF Karaoke Message

- This is special banner sequence data targeted for displaying banners such as for karaoke.
 - Lyrics banner data can be “wiped” in synchronization with the song.
 - Karaoke messages can be broadly divided into lyrics data and control cue data.
 - Karaoke messages can be given in the following three ways, and handling is up to the processing system.
 - 1) Specified as a meta-event within the track chunk.
 - 2) Specified in meta-event format, in a separate chunk following the track chunk.
 - Chunk type: XFKM
 - Data length specifier: 4 bytes
 - <XFKM Event> = <delta-time> <Meta Event>
 - 3) Specified with the chunk of 2) placed in a separate file.
 - The contents of the file are identical to chunk 2).
 - DOS filename extension: .XKM
 - Mac file type: XFKM
- If two or more of these exist, they will take priority in the order of 3), 2), 1).

<Lyrics data>

- This defines the specifications of the lyrics information of a song, and provides sufficient data to create lyric sheets and karaoke applications, etc.

Lyrics

- The Lyric meta-event defined by SMF is used without change as the data format.
- For the character code that is used, refer to “Appendix A. Character codes.”
- The character code that is used is specified within the XF Lyric header that precedes the lyric events. For details refer to 1. Contents at the beginning of the track chunk.
- It is desirable that the timing data of the lyrics events have the same value as the notes of the melody. However as long as major discrepancies do not occur in the wipe/display timing, this can move forward or backward.

FF 05 len <Text>

Control characters

Lyrics data can include control characters for performing various types of control.

ASCII (single-byte) is used for control characters.

If you wish to display the control character itself with the lyrics, prefix it with a single-byte backslash “\” control character.

FF 05 len <text>

1) “^” Space

This is used in place of a space (blank code).

The width of spacing is left up to the display system.

Example:

Wind | blows | through | the | trees^ | The | moon | bounces | off | the | water |

Display output

Wind blows through the trees The moon bounces off the water

2) “/” Carriage return

This is used instead of a carriage return code.

Insert it at a position where you want to start a new line in the display.

Example:

|If |music |be |the |food |of |love,/|play |on.|

Display output

If music be the food of love,
play on.

If the note corresponding to the character string that precedes the carriage return has a long note value, it is desirable to insert the carriage return as an independent Lyric meta-event at the key-off timing, so that when using a “wipe” you can adjust the timing at which the wipe ends.

Example:

So it goes (the note “goes” has a long duration)

× |So |it |goes/|

○ |So |it |goes|/|

3) “%” Line feed

This is used to indicate semantic breaks of the text within a single line.

This is used to allow natural line breaks even on low-resolution display systems.

4) “<” Page break

This is added to the beginning of the character string that you want to appear as the first line, so that the lyrics will be displayed starting on a new page.

The display system shall ensure that the character string that follows “<” will be displayed on the first line.

Example:

|<Shall |I |compare |thee |

|to |a |summer’s |day? /|

|Nay, |thou |art |fairer/|

Display output

Shall I compare thee to a summer’s day?
Nay, thou art fairer

Add the page break to the beginning of the character string that you wish to display as the first line.

Example:

× |<|Shall |I |compare |thee |to |a |summer’s |day?|

○ |<Shall |I |compare |thee |to |a |summer’s |day?|

5) “>” Tab

This indicates a horizontal tab.

The width of the spacing produced by the tab is left up to the display system.

A tab can be input only at the beginning of a line.

Example:

|<Shall |I |compare |thee/|>To |a |summer’s |day?|

Display output

Shall I compare thee
To a summer’s day?

6) “\” Backslash

“\” backslash causes the following single-byte ASCII character to lose its function as a control character, and be treated as a displayed character.

Prefix a backslash to a control character when you wish to display that control character itself within the lyrics.

<Cue data>

- The following text data adds various functionality to the processing system.
- These use the SMF-defined meta-event Cue Point.
- These are written in ASCII.

FF 07 len <Text>

1) XF Lyrics Header

Information on the subsequent Lyrics meta-events.

For details refer to 1. Contents at the beginning of the track chunk.

Example:

FF 07 len <\$Lyr:4,12:240:US>

2) Vocal Part Cues

These indicate the part and type of the lyrics.

They begin with an ampersand “&”.

Male	&m
Female	&f
Chorus	&c
Solo	&s
Mixed	&p
Spoken line	&w
Non-lyric	&x

The “non-lyric” part cue is used to display messages other than lyrics.

For example to display “Interlude” or “Introduction,” you would insert the “Non-lyric” part cue, and then use Lyric events to specify “Interlude” or “Introduction.”

When a “Spoken line” or “Non-lyric” cue event occurs, the processing system shall not wipe Lyric meta-events until the next cue event other than “Spoken line” or “Non-lyric” occurs.

Example:

Interlude

FF 07 len <&x>

FF 05 len <Interlude>

5. Style Messages

- These are used to define chords, styles, and style sections etc.
- These are specified in the form of Sequencer-Specific meta-events.
- The addition of Style Messages to the XF format shall be considered as XF format version 2.0

[Key Signature]

- The SMF Key Signature meta-event is used for key signature data.

[Chord Name]

- Specify the chord name.

FF 7F 07 43 7B 01 cr ct bn bt Chord Name

cr Chord root Offnnnnn

fff: Inversion number, nnnn: Note

0000nnnn	0x	bbb (three flats)
0010nnnn	1x	bb (two flats)
0001nnnn	2x	b (one flat)
0011nnnn	3x	natural (natural)
0100nnnn	4x	# (one sharp)
0101nnnn	5x	## (two sharps)
0110nnnn	6x	###(three sharps)
0fff0000	x0	reserved (not used)
0fff0001	x1	C
0fff0010	x2	D
0fff0011	x3	E
0fff0100	x4	F
0fff0101	x5	G
0fff0110	x6	A
0fff0111	x7	B

§2 MIDI Specifications

ct	Chord type	0 - 34	
00000000	00	0	Maj
00000001	01	1	Maj6
00000010	02	2	Maj7
00000011	03	3	Maj7(#11)
00000100	04	4	Maj(9)
00000101	05	5	Maj7(9)
00000110	06	6	Maj6(9)
00000111	07	7	aug
00001000	08	8	min
00001001	09	9	min6
00001010	0A	10	min7
00001011	0B	11	min7b5
00001100	0C	12	min(9)
00001101	0D	13	min7(9)
00001110	0E	14	min7(11)
00001111	0F	15	minMaj7
00010000	10	16	minMaj7(9)
00010001	11	17	dim
00010010	12	18	dim7
00010011	13	19	7th
00010100	14	20	7sus4
00010101	15	21	7b5
00010110	16	22	7(9)
00010111	17	23	7(#11)
00011000	18	24	7(13)
00011001	19	25	7(b9)
00011010	1A	26	7(b13)
00011011	1B	27	7(#9)
00011100	1C	28	Maj7aug
00011101	1D	29	7aug
00011110	1E	30	1+8
00011111	1F	31	1+5
00100000	20	32	sus4
00100001	21	33	1+2+5
00100010	22	34	cc
bn	On-bass note		Same as Chord root, 127:No bass chord
bt	Bass chord		Same as Chord type 127:No bass chord

[Rehearsal Mark]

FF 7F 04 43 7B 02 rr Rehearsal Mark

rr Rehearsal Mark 0yyyxxxx

Lower 4 bits (xxxx)

0: Intro
 1: Ending
 2: Fill-in
 3: A
 4: B
 :
 15: M

Upper 3 bits (yyy)

0-7: Variation
 1: the conventional 'A', 'B' etc.
 2: the conventional "A", "B" etc.
 :

- These are used to specify section divisions.
- Place them at the beginning of a measure.
- It is possible for there to be a song without some rehearsal marks.
- The number of measures between rehearsal marks is not prescribed.
- This is to be distinguished from the YMCS-defined Section Control system exclusive (F0,43,7E,00,ss,dd,F7).
- Although these are being introduced for the sake of Style functionality, they may of course be used for score display etc. These may be used as desired.

[Phrase mark]

FF 7F 05 43 7B 03 xx yy Phrase mark

xx Phrase mark 0hfcccc

bit6(h)

0:Right, 1:Left right hand / left hand

bit5(f)

0:CH enabled, 1:CH disabled Channel disabling flag
 Enabled/disabled condition of the bit 4-0 channel data
 For a phrase common to all channels, this is 1

bit4-0(ccccc)

CH(0:1ch - 31:32ch) Channel data for which the phrase mark is valid

yy Phrase mark level 1-127

0: Reserved
 1: Level 1 (guide phrase level)
 2: Level 2
 :
 8: Level 8 (DOC phrase level)
 :

Guidelines for phrase mark levels

Level 1 (Guide Phrase Level)

1. This level defines the shortest practice unit that is used when practicing a composition.
2. The number of notes in a single phrase should be from several notes to a dozen or more, and at most shall be no more than 32 notes.
3. This is used basically for motifs of 1–2 measures, but may be used to divide shorter phrases for practice if desired.
4. This phrase mark shall be inserted before the first key-on of the track into which it is inserted, if possible at a timing of 0.
5. Subsequently, this phrase mark is inserted at each short practice unit. One of these phrase marks must also be inserted after the last key-on of the track.
6. Level 1 shall be included within level 8.

Level 2 (Motif divisions)

1. This is used when it is necessary to divide the music into performance motifs as the smallest unit.
2. This is close to the phrase slurs used in piano music etc.
3. The purpose of this division is to precisely extract elements of performance technique.
4. These may be used as divisions for an individual channel.
5. These may not overlap level 3 divisions.

Levels 3 - 7 (Practice divisions)

1. These are used to indicate divisions for practicing.
2. The length of a unit is not prescribed. (Normally, level 3 will be approximately four measures, and level 4 will be approximately eight measures.)
3. These may be used as divisions for an individual channel.
4. These may not overlap a higher-level division.
5. Use these starting from the lower level. It is not necessary to make settings for all levels.

Level 8 (paragraph divisions)

1. These shall be compatible with DOC phrase marks.
2. Use these to divide at musical paragraphs where an unnatural impression will not be produced when all tracks are played back.
3. For simple forms such as popular songs, these coincide with divisions of the “A” melody and “B” melody, etc.
4. These are all-channel divisions.
5. These divisions may not cross level 9 divisions.

Level 9 (chorus divisions)

1. These create large divisions such as between the intro, first chorus, second chorus, and ending etc.
2. If the composition is not structured by verses, these correspond to divisions such as “theme” and “development” etc.
3. Insert this division at the first note that follows the setup data or other silent portion at the beginning of the data.
4. These are all-channel divisions.

Note: Inclusion of level divisions

For divisions of any level whose channel specifiers match, upper levels shall surround lower levels. (Thus, level 9 divisions must always surround level 8 divisions.)

[Max Phrase mark]

FF 7F 04 43 7B 04 xx Max Phrase mark

xx Maximum number of phrases 1-127

Input a number that is one less than the number of Level 8 phrase marks you input.

[Guide Track Flag]

- This event is used by the processing system to determine whether the melody is only on channel 1, or divided into channel 1 (right hand) and channel 2 (left hand).
- Use this event to specify the channel(s) that contain the melody.

FF 7F 05 43 7B 0C rr ll Guide Track Flag

rr Melody Track 1 Ch (right hand) 0:none, 1:1ch..16:16ch

ll Melody Track 2 Ch (left hand) 0:none, 1:1ch..16:16ch

- The basic setting is rr=01H, ll=02H.
- This event must be input at a timing of 0.
- If this event does not exist, it will be assumed that channel 1 is the only melody track.

[Information flag for Gt (guitar information flags)]

- These events are used by the processing system to calculate string information and fret locations for the guitar part.
- Data length is variable, and depends on the number of strings.

FF 7F len 43 7B 10 xx pp kk aa bb cc dd ee ff

xx Applicable channel 00fccccc

f 0 : Applicable channel data in bits 4-0 is valid

1 : All channels common (applicable channel data in bits 4-0 is invalid)

ccccc 0: 1ch - 31: 32chl

Indicates the channel of the first string

This channel and following channels up to the number of strings are specified as the guitar part.

(skip ch.10)

11ch (0 x 0 A) is usual

pp Applicable part

0: Guitar

1: Bass

2: Ukulele

Other: Reserve

kk Capo location data 0 - 23

aa First string note no. 0-127
 bb Second string note no. 0-127
 cc Third string note no. 0-127
 dd Fourth string note no. 0-127
 ee Fifth string note no. 0-127
 ff Sixth string note no. 0-127

This data is placed at a timing of 0.

[Chord Voicing for Gt (chord voicing for guitar)]

FF 7F len 43 7B 12 xx aa AA bb BB cc CC ...

xx Applicable channel 00fcccc

f 0 : Applicable channel data in bits 4-0 is valid
 1 : All channels common (applicable channel data in bits 4-0 is invalid)

cccc Applicable channel

Specify the channel specified by the Guitar Information Flag.
 (Necessary if there is data for two or more guitars = when information flag for Gt is 2 or higher)

aa First string fret position 000yyyyy

0 - 24
 0: Open string
 127: Unplayed string

AA First string fingering 00000fff

0: (No finger number) Not pressed
 1 - 5: Finger number 1: Thumb

bb Second string fret position 00000fff

0 - 24
 0: Open string
 127: Unplayed string

BB First string fingering 00000fff

0: (No finger number) Not pressed
 1 - 5: Finger number 1: Thumb
 0: (No finger number) Not pressed

Data Length is variable, and depends on the number of strings.

[Fingering Number]

FF 7F 06 43 7B 05 cc nn ff

Fingering Number

cc Ch. of applicable note 0xxxxccc

Lower 5 bits (ccccc) Channel number of the applicable note

0: 1ch - 31: 32ch

Upper 2 bits (xx)

00: Normal keyboard instrument fingering

01: Multi-channel fingering (for guitar fingering)

In this case, the channel specified by the Guitar Information Flag is input to the lower 5 bits.

The applicable note name is ignored.

Other: reserved

nn Applicable note name 0 - 127

ff Fingering number ff 0yyyzfff

Lower 3 bits (fff)

Use 1-5 to specify the fingering number

Middle 1 bit (z)

0: Right (right hand), 1: Left (left hand)

Upper 3 bits (yyy)

000: Normal keyboard instrument fingering

Other: reserved

- Currently, the fingering numbers for playing keyboard instruments are defined.
- Special performance technique data is undefined (reserved).
- Data other than for keyboard instruments is undefined (reserved).

Appendix A. Character codes

The following table defines the symbols used to specify the character code used in the XG information header or in lyric data.

Symbol: Symbols used as data items in the XF information header or XG lyric header to specify the character conversion code.

Character conversion code: Character conversion code indicated by "Symbols."

Supported languages: Languages that can be expressed using the applicable character conversion code.

Symbol	Character code	Supported languages
L1	Latin 1 (ASCII(7bit) + ISO 8859-1)	English, French, German, Italian, Spanish, Portuguese, etc.
JP	Shift-JIS	Japanese
KR	ISO-2022-KR (ASCII + KS C 5601-1992 incl. Johab)	Korean
HZ	HZ-GB-2312	Chinese (Simplified Chinese)
B5	Big Five	Chinese (Traditional Chinese) (including Taiwanese etc.)
CY	KOI8-R	Russian etc.
VN	TCVN-5773:1993	Vietnamese

Appendix B. Country codes

The following two-character codes are used to indicate the “Country of production” data item in the XF information header (common).

AD:Andorra, **AE**:United Arab Emirates, **AF**:Afghanistan, **AG**:Antigua and Barbuda,
AI:Anguilla, **AL**:Albania, **AM**:Armenia, **AN**:Netherlands Antilles, **AO**:Angola, **AQ**:Antarctica,
AR: Argentina, **AS**: American Samoa, **AT**:Austria, **AU**:Australia, **AW**:Aruba, **AZ**:Azerbaijan,
BA:Bosnia-Herzegovina, **BB**:Barbados, **BD**:Bangladesh, **BE**:Belgium, **BF**:Burkina Faso,
BG:Bulgaria, **BH**:Bahrain, **BI**:Burundi, **BJ**:Benin, **BM**:Bermuda, **BN**:Brunei Darussalam,
BO:Bolivia, **BR**:Brazil, **BS**:Bahamas, **BT**:Bhutan, **BV**:Bouvet Island, **BW**:Botswana,
BY:Belarus, **BZ**:Belize,
CA:Canada, **CC**:Cocos (Keeling) Islands, **CF**:Central African Republic, **CG**:Congo,
CH:Switzerland, **CI**:Cote d’Ivoire, **CK**:Cook Islands, **CL**:Chile, **CM**:Cameroon, **CN**:China,
CO:Colombia, **CR**:Costa Rica, **CU**:Cuba, **CV**:Cape Verde, **CX**:Christmas Island, **CY**:Cyprus,
CZ:Czech Republic,
DE:Germany, **DJ**:Djibouti, **DK**:Denmark, **DM**:Dominica, **DO**:Dominican Republic,
DZ:Algeria,
EC:Ecuador, **EE**:Estonia, **EG**:Egypt, **EH**:Western Sahara, **ES**:Spain, **ET**:Ethiopia,
FI:Finland, **FJ**:Fiji, **FK**:Falkland Islands, **FM**:Micronesia, **FO**:Faroe Islands, **FR**:France,
GA:Gabon, **GD**:Grenada, **GE**:Georgia, **GF**:French Guiana, **GH**:Ghana, **GI**:Gibraltar,
GL:Greenland, **GM**:Gambia, **GN**:Guinea, **GP**:Guadeloupe, **GQ**:Equatorial Guinea,
GR:Greece, **GT**:Guatemala, **GU**:Guam, **GW**:Guinea-Bissau, **GY**:Guyana,
HK:Hong Kong, **HM**:Heard and McDonald Islands, **HT**:Haiti, **HU**:Hungary,
ID:Indonesia, **IE**:Ireland, **IL**:Israel, **IN**:India, **IO**:British Indian Ocean Territory, **IQ**:Iraq,
IR:Iran, **IS**:Iceland, **IT**:Italy,
JM:Jamaica, **JO**:Jordan, **JP**:Japan,
KE:Kenya, **KG**:Kyrgyzstan, **KH**:Cambodia, **KI**:Kiribati, **KM**:Comoros,
KN:Saint Kitts and Nevis, **KR**:Korea, **KW**:Kuwait, **KY**:Cayman Islands, **KZ**:Kazakhstan,
LA:Lao People’s Democratic Republic, **LB**:Lebanon, **LC**:Saint Lucia, **LI**:Liechtenstein,
LK:Sri Lanka, **LR**:Liberia, **LS**:Lesotho, **LT**:Lithuania, **LU**:Luxembourg, **LV**:Latvia,
LY:Libyan Arab Jamahiriya,
MA:Morocco, **MC**:Monaco, **MD**:Moldova, **MG**:Madagascar, **MH**:Marshall Islands, **ML**:Mali,
MM:Myanmar, **MN**:Mongolia, **MO**:Macau, **MP**:Northern Mariana Islands, **MQ**:Martinique,
MR:Mauritania, **MS**:Montserrat, **MT**:Malta, **MU**:Mauritius, **MV**:Maldives, **MW**:Malawi,
MX:Mexico, **MY**:Malasia, **MZ**:Mozambique,
NA:Namibia, **NC**:New Caledonia, **NE**:Niger, **NF**:Norfolk Island, **NG**:Nigeria, **NI**:Nicaragua,
NL:Netherlands, **NO**:Norway, **NP**:Nepal, **NR**:Nauru,
NT:Neutral Zone (between Saudi Arabia & Iraq), **NU**:Niue, **NZ**:New Zealand,
OM:Oman,
PA:Panama, **PE**:Peru, **PF**:French Polynesia, **PG**:Papua New Guinea, **PH**:Philippines,
PK:Pakistan, **PL**:Poland, **PM**:Saint Pierre and Miquelon, **PN**:Pitcairn, **PR**:Puerto Rico,
PT:Portugal, **PW**:Palau, **PY**:Paraguay,
QA:Qatar,
RE:Re’union, **RO**:Romania, **RU**:Russian Federation, **RW**:Rwanda,
SA:Saudi Arabia, **SB**:Solomon Islands, **SC**:Seychelles, **SD**:Sudan, **SE**:Sweden,
SG:Singapore, **SH**:Saint Helena, **SI**:Slovenia, **SJ**:Svalbard and Jan Mayen islands,
SK:Slovakia, **SL**:Sierra Leone, **SM**:San Marino, **SN**:Senegal, **SO**:Somalia, **SR**:Suriname,
ST:Sao Tome and Principe, **SU**:Former Soviet Union, **SV**:El Salvador, **SY**:Syria,
SZ:Swaziland,
TC:Turks and Caicos Islands, **TD**:Chad, **TF**:French Southern Territories, **TG**:Togo,
TH:Thailand, **TJ**:Tajikistan, **TK**:Tokelau, **TM**:Turkmenistan, **TN**:Tunisia, **TO**:Tonga,
TP:East Timor, **TR**:Turkey, **TT**:Trinidad and Tobago, **TV**:Tuvalu, **TW**:Taiwan, **TZ**:Tanzania,
UA:Ukraine, **UG**:Uganda, **UK**:United Kingdom, **UM**:United States Minor Outlying Islands,
US:United States, **UY**:Uruguay, **UZ**:Uzbekistan,

§2 MIDI Specifications

VA:Vatican City State, **VC:**Saint Vincent and the Grenadines, **VE:**Venezuela,
VG:Virgin Islands (British), **VI:**Virgin Islands (U.S.), **VN:**Vietnam, **VU:**Vanuatu,
WF:Wallis & Futuna Islands, **WS:**Samoa,
YE:Yemen, **YU:**Yugoslavia,
ZA:South Africa, **ZM:**Zambia, **ZR:**Zaire, **ZW:**Zimbabwe

Appendix C. Song genre

The following words shall be used to indicate the “Song genre (Category)” data item in the XF information header (common).

Ballroom	music for social dancing
Blues	blues
Bossa-Samba	bossanova, samba
Children’s Song	children’s songs and nursery songs
Christmas-Hymn	Christmas music, hymns
Classical	classical music
C&W	country, western
Dance	dance music, disco
Easy Listening	easy listening
Folk	folk songs
Jazz-Fusion	jazz, fusion
J.Enka	Japanese “enka”
J.Folk-GS	Japanese folk songs, group sounds
J.Pop	Japanese pop
J.Kayoukyoku	Japanese “kayoukyoku”
J.Minyou	Japanese “minyou”
Latin	Latin music
March	march
NewAge	new age music
Reggae	reggae
Rock	rock
Rock’n Roll-R&B	rock’n’roll, rhythm & blues
Pops	pops
Soul-Funk	soul music, funk
Soundtrack	film music
Standard	standards
Techno	techno
World	world music, ethnic music
Others	other music

Appendix D. Song keywords

The following words shall be used to specify the “Song keyword (Keyword)” data item in the XF information header (common).

movie
TV
Dramatic
Love song
Old Familiar Melody
Contemporary
Ballad

Calm
Rhythmical
Cheerful
Joy
Humorous
Sad
Scary
Horror
Dangerous
High-spirited
Grand
Epic
Passionate
Intense
Surprise
Cool
Serious
Good mood
Mysterious
Ethnic

Friends
Love
Wedding
Parting
Lost Love
One-sided Love
Tears
Liquor
Dream

Natural
Sea
Sky
Spring
Summer
Autumn
Winter
Snow
Rain
Christmas

Appendix E. XG Song Data Number

- This is data used to manage Yamaha XG song data.
- This is a unique management number used within Yamaha for purposes of royalty transactions, payment of copyright fees, collection of online sale fees, and other management duties.

FF 7F 0D 43 7B 7F c1 c2 c3 c4 c5 c6 c7 c8 v1 v2

c1..c8	song data ID
v1,v2	song data version No.