

# SMF with Lyrics Data Creation Guideline

September 4, 1998

## 1. Objectives

Lyrics are as extremely important as instrumental parts in a music piece. However, in current SMF features, the Lyric Meta Event processing is designed for English speaking environments. This causes some problems, for example, there is no regulation for multi-byte character code such as Japanese. Also, the lack of a precise standard in displaying lyrics makes it difficult to have a data compatibility among different applications used for replaying Karaoke or displaying lyrics with music notes.

SMF with Lyrics is designed to be expanded as Lyric/Display Meta Event in the form of SMF (Standard MIDI File) that is a market standard format. It enables to respond to multi-byte character codes and also some rules for displaying lyrics and song information. This expanded SMF is available to display not only English but also Japanese characters for music scores or Karaoke. It is also able to deal with song information such as song titles, composer's names, songwriter's names and artist's names that are also useful for Karaoke function. Adding Ruby is also available on lyrics and song information. As SMF with lyrics is based on SMF, it has a great usability and compatibility.

This guideline is designed to provide necessary information to create data by applying new rules described in SMF with Lyrics including Ruby, Character Code information and Song information. It will help data creators.

## 2. The items to be described with Lyric/Display Meta Event

Lyric/Display Meta Event is one of the events to work for text type information. It mainly describes about display information.

- Character Code Information
- Song information as shown below:
  - Song title, original song title
  - Composer's name
  - Song writer's name
  - Artist's name, Performer's name
- Ruby
- Information for display such as layout (control character):
  - Carriage Return ( shown as CR )
  - Line Feed ( LF )
  - Horizontal Tab ( HT )
- Lyrics

Please do not use Lyric/Display Meta Event for other purposes except the cases listed above.

### 3 Specific grammar for Lyric/Display Meta Event

#### 3-1. Character Code Information

This is the information to set a character code set to display contents of Lyric/Display Meta Event. Character Code Information should be placed at the top of Lyric/Display Meta Event in SMF. This information is effective until next Character Code Information comes.

Character Code Information for Japanese Character display:

MS-Kanji (Shift-JIS) character code set should be assigned.

```
{@JP} or {@Jp} or {@jp}
```

Example)

```
1:1:000<Meta>Lyric/Display "{@JP}"
```

Ordinary European language character code information:  
ANSI Character code set should be assigned.

```
{@LATIN} or {@Latin} or {@latin}
```

Example)

```
1:1:000<Meta>Lyric/Display "{@LATIN}"
```

Character code information is always placed from the top of the event and should be typed as one event. Please do not type it by dividing the information into more than one event.

Use ASCII code (1 byte character) for Character code information. 2-byte character is not possible to use.

If there is no Character code information, it is considered that ANSI character set is assigned.

Character sets, except the sets listed above, are not defined yet. There is a possibility that other sets will be newly defined by MMA or AMEI in the future. Please do not define new character set without discussing with MMA or AMEI.

There is a possibility that no Lyric/Display Meta Event will be displayed until proper character code shows up if undefined character information or incomplete character code information is typed. Pay extra attention to type character code information precisely.

#### 3-2. Song Information

Songwriter's name, composer's name, artist's name and song title are able to display by using Lyric/Display Meta Event. When you want to use Japanese characters, you have to assign MS-Kanji (Shift-JIS) code set for character code information first, and then type song information.

There are four items for song information. You don't have to type them all. Please type items you need only.

When you finish typing all the song information, type "{#}"(Null Tag) to show that it is the end of the information.

Song title/original title:

{#TITLE=} or {#Title=} or {#title=}

Composer's name:

{#COMPOSER=} or {#Composer=} or {#composer=}

Song writer's name:

{#LYRICS=} or {#Lyrics=} or {#lyrics=}

Artist's name/performer's name:

{#ARTIST=} or {#Artist=} or {#artist=}

The end of song information:

{#}

Example)

```
1:1:000<Meta>Lyric/Display "{@JP}"
1:1:000<Meta>Lyric/Display "{#TITLE=イエスタデイ・ワンス・モア}"
1:1:000<Meta>Lyric/Display "{#COMPOSER=リチャード・カーペンター}"
1:1:000<Meta>Lyric/Display "{#LYRICS=ジョン・ベティス}"
1:1:000<Meta>Lyric/Display "{#ARTIST=カーペンターズ}"
1:1:000<Meta>Lyric/Display "{#}"
```

All the items of song information should be typed from the top of the event. Please avoid combining more than one item as one event when you type them.

Example) for a common mistake)

```
× 1:1:000<Meta>Lyric/Display "{#TITLE=ミッシェル}{#ARTIST=ビートルズ}"
```

Song information can be divided into more than one Lyric/Display Meta Event when it is typed.

You can change character code information in the middle of song information.

Example)

Song Title: "Étoile d'été ~なつのせいざ~"

```
1:1:000<Meta>Lyric/Display "{@LATIN}"
1:1:000<Meta>Lyric/Display "{#TITLE=Étoile d'été}"
1:1:000<Meta>Lyric/Display "{@JP}"
1:1:000<Meta>Lyric/Display "~なつのせいざ~"
1:1:000<Meta>Lyric/Display "{#}"
```

You can also use layout information (CR, HT etc) in the middle of the song information.

Example)

(display)

歌劇「蝶々夫人」

アリア「ある晴れた日に」

(description)

```
1:1:000<Meta>Lyric/Display "{@JP}"
1:1:000<Meta>Lyric/Display "{#TITLE=歌劇「蝶々夫人」}"
1:1:000<Meta>Lyric/Display "\r"
1:1:000<Meta>Lyric/Display "\t"
1:1:000<Meta>Lyric/Display "アリア「ある晴れた日に」}"
1:1:000<Meta>Lyric/Display "{#}"
```

### 3-3. Ruby

#### 3-3-1. Basic use of Ruby

You can select an area of information to display as Ruby by using Lyric/Display Meta Event. We recommend applying Ruby for all the Kanji characters or symbols in Lyric/Display Meta Event to prepare for the case when only kana is displayed.

To type Ruby, please type letters that you want to use as Ruby by using "[" and "]" (in 1 byte character) right after the characters you want to apply Ruby. The letters between "[" and "]" will add Ruby on the letters of the Lyric/Display Meta Event typed right before "[".

Example)

```
18:1:000<Meta>Lyric/Display "夜[よ]"
25:3:000<Meta>Lyric/Display "恋[こい]"
```

Ruby is available not only for Kanji characters but symbols.

Example)

```
30:1:000<Meta>Lyric/Display "◎[にじゅうまる]"
```

We recommend to make the character set to add Ruby and the character set of Ruby itself as one lyric/Meta Event. If there is no character to be Ruby right before "[", the previous Lyric/Display Meta Event will be considered as characters to add Ruby on.

If there are more than one character between "[" and "]" , and if you desire to have Ruby in different timing, you can divide them into multi events.

Example)

```
25:3:000<Meta>Lyric/Display "恋[こ"
25:4:000<Meta>Lyric/Display "い]"
```

Not recommended Example)

```
25:3:000<Meta>Lyric/Display "恋"
25:3:000<Meta>Lyric/Display "[こ"
25:4:000<Meta>Lyric/Display "い]"
```

When those characters are meant for one word, Ruby should be applied on each Kanji basically.

Example)

```
21:1:000<Meta>Lyric/Display "地[ち]"
21:2:000<Meta>Lyric/Display "球[きゅう]"
```

When there is a certain way to read a word with more than one Kanji, Ruby should be typed without being separated.

Example)

```
24:2:000<Meta>Lyric/Display "今日[きよ]"
24:3:000<Meta>Lyric/Display "う]"
```

Example)

```
40:2:000<Meta>Lyric/Display "明日[あ]"
40:2:048<Meta>Lyric/Display "し]"
40:3:000<Meta>Lyric/Display "た]"
```

Inappropriate Example)

```
× 40:2:000<Meta>Lyric/Display "明[あ]"
× 40:2:048<Meta>Lyric/Display "し]"
× 40:3:000<Meta>Lyric/Display "日[た]"
```

Please apply the same way when you want to have special Ruby on a certain character set.

Example)

```
42:3:000<Meta>Lyric/Display "世間[ひ]"
42:4:000<Meta>Lyric/Display "と]"

44:1:000<Meta>Lyric/Display "電腦機械[コン]"
44:2:000<Meta>Lyric/Display "ピュー]"
44:3:000<Meta>Lyric/Display "ター]"

45:2:000<Meta>Lyric/Display "感謝祭[イ]"
45:3:000<Meta>Lyric/Display "ー]"
45:3:048<Meta>Lyric/Display "ス]"
45:4:000<Meta>Lyric/Display "ター]"
```

You can add Ruby on 1 byte character, too. (Please refer 4-3)

Example)

```
44:4:000<Meta>Lyric/Display "{@JP}"
45:1:000<Meta>Lyric/Display "my[マイ]"
45:1:000<Meta>Lyric/Display "home[ホーム]"
```

When a character with no Ruby and a character with Ruby are combined, you can make it as one event by typing empty Ruby. Empty Ruby, "[ ]", will be used to make sure where you want to apply Ruby.

Example)

```
1:1:000<Meta>Lyric/Display "{#TITLE=青[あお]い[ ]鳥[とり]}"
```

If you did not type empty Ruby, "とり" will become Ruby for "い鳥".

Example of a mistake)

```
× 1:1:000<Meta>Lyric/Display "{#TITLE=青[あお]い鳥[とり]}"
```

When you type more than one event for a word, you don't have to type empty Ruby.

Example)

```
1:1:000<Meta>Lyric/Display "{#TITLE=動[どう]物[ぶつ]"
```

```
1:1:000<Meta>Lyric/Display "の"
```

```
1:1:000<Meta>Lyric/Display "謝[しゃ]肉[にく]祭[さい]}"
```

You can use Ruby not only for lyrics but also for song information.

Example)

```
1:1:000<Meta>Lyric/Display "{#TITLE=青[あお]い[ ]鳥[とり]}"
```

```
1:1:000<Meta>Lyric/Display "{#COMPOSER=渡[わた]辺[なべ]よう[ ]子[こ]}"
```

```
1:1:000<Meta>Lyric/Display "{#LYRICS=山[やま]田[だ]昭[しょう]ー[い  
ち]}"
```

```
1:1:000<Meta>Lyric/Display "{#}"
```

### 3-3-2. Exceptional use of Ruby

To switch character codes in Ruby area, you have to finish Ruby once right before the new character code and make character code information as a different event and type Ruby after that.

When there is no character to add Ruby between Ruby and another Ruby, those Rubies is shown as one Ruby.

Example) Switch character codes from "{@JP}" to "{@LATIN}" in Ruby.

When you want to type Ruby "わたしの amour" on "恋しい人":

```
1:1:000<Meta>Lyric/Display "{@JP}"
```

```
1:1:000<Meta>Lyric/Display "恋しい人[わたしの]"
```

```
1:1:000<Meta>Lyric/Display "{@LATIN}[amour]"
```

"わたしの" and "amour" are shown as one Ruby "わたしの amour".

To put character code in Ruby is prohibited.

Example of a mistake)

```
× 1:1:000<Meta>Lyric/Display "{@JP}"
× 1:1:000<Meta>Lyric/Display "恋しい人[わたしの {@LATIN}amour]"
```

Example of a mistake)

```
× 1:1:000<Meta>Lyric/Display "{@JP}"
× 1:1:000<Meta>Lyric/Display "恋しい人[わたしの "
× 1:1:000<Meta>Lyric/Display "{@LATIN}amour]"
```

Example of a mistake)

```
× 1:1:000<Meta>Lyric/Display "{@JP}"
× 1:1:000<Meta>Lyric/Display "恋しい人[わたしの "
× 1:1:000<Meta>Lyric/Display "{@LATIN}"
× 1:1:000<Meta>Lyric/Display "amour]"
```

The same rule is applied when you type Ruby on song information.

Example)

```
1:1:000<Meta>Lyric/Display "{@LATIN}"
1:1:000<Meta>Lyric/Display "{#ARTIST=Nat Cole
1:1:000<Meta>Lyric/Display "{@JP}"
1:1:000<Meta>Lyric/Display "[ナット・コール]"
```

Example of a mistake)

```
× 1:1:000<Meta>Lyric/Display "{@LATIN}"
× 1:1:000<Meta>Lyric/Display "{#ARTIST=Nat Cole[ {@JP}ナット・コー
ル]}"
```

Example of a mistake)

```
× 1:1:000<Meta>Lyric/Display "{@LATIN}"
× 1:1:000<Meta>Lyric/Display "{#ARTIST=Nat Cole["
× 1:1:000<Meta>Lyric/Display "{@JP}ナット・コール]"
```

To put song information in Ruby is prohibited.

Example of a mistake)

```
× 1:1:000<Meta>Lyric/Display "{@JP}"
× 1:1:000<Meta>Lyric/Display "{#TITLE=Yesterday[イエスタデイ・"
× 1:1:000<Meta>Lyric/Display "{#ARTIST=ビートルズの}}}"
× 1:1:000<Meta>Lyric/Display "{#"
```

### 3-4. Control Character

#### 3-4-1. Basic use of Control character

Carriage return CR (0D):

This is used to show the end of the line. This is placed after the Lyric/Display Meta Event of the end of the line. Also, it is used at the end of the sentence or when a meaning of the connected words is desired to be ending there. Carriage Return is an independent Lyric/Display Meta Event that only

works as CR.

Line Field LF (0A):

It is used to divide paragraphs. This is often used between the first verse and the second verse of a song and show bigger gap between them. Line Field is an independent Lyric/Display Meta Event which only works as LF.

Horizontal Tab HT (09):

Horizontal Tab is an independent Lyric/Display Meta Event that only works as HT.

When you can not use control characters ( CR、 LF、 HT ) , you can get the same result by combining backslash "\ " and characters. This is considered as an independent event that works as control characters.

Typing:	Meaning:
\r	Carriage Return (0D)
\n	Line Field (0A)
\t	Horizontal Tab (09)

Example)

```
20:3:000<Meta>Lyric/Display "だ"  
21:1:000<Meta>Lyric/Display "よ"  
21:4:000<Meta>Lyric/Display 0D  
24:3:000<Meta>Lyric/Display "だ"  
25:1:000<Meta>Lyric/Display "ね"  
25:4:000<Meta>Lyric/Display 0D  
25:4:005<Meta>Lyric/Display 0A
```

Example)

```
20:3:000<Meta>Lyric/Display "だ"  
21:1:000<Meta>Lyric/Display "よ"  
21:4:000<Meta>Lyric/Display "\r"  
24:3:000<Meta>Lyric/Display "だ"  
25:1:000<Meta>Lyric/Display "ね"  
25:4:000<Meta>Lyric/Display "\r"  
25:4:005<Meta>Lyric/Display "\n"
```

### 3-4-2. Control characters in Ruby

Control characters are not subjected to be shown with Ruby. Please do not out control characters or symbols which work as the same as control characters in the middle of the characters that you need Ruby on or in the middle of Ruby itself.

Example of a mistake)

```
× 1:1:000<Meta>Lyric/Display "今夜\r[こんや]"
```

Example of a mistake)

```
× 1:1:000<Meta>Lyric/Display "今夜\r"  
× 1:1:000<Meta>Lyric/Display "[こんや]"
```



Example of a mistake)

```
× 1:1:000<Meta>Lyric/Display "今夜"  
× 1:1:000<Meta>Lyric/Display "\r"  
× 1:1:000<Meta>Lyric/Display "[こんや]"
```

Example of a mistake)

```
× 1:1:000<Meta>Lyric/Display "今夜[こん\tや]"
```

Example of a mistake)

```
× 1:1:000<Meta>Lyric/Display "今夜[こん"  
× 1:1:000<Meta>Lyric/Display "\t"  
× 1:1:000<Meta>Lyric/Display "や]"
```

Example of a mistake)

```
× 1:1:000<Meta>Lyric/Display "今夜[こん]"  
× 1:1:000<Meta>Lyric/Display "\t"  
× 1:1:000<Meta>Lyric/Display "[や]"
```

### 3-5. Characters that require a special typing...

Because a back slash, "\" (5C), "{" (7B), "}" (7D), "[" (5B), "]" (5D) ---you see ASCII code number in each pair of“( )” ---, is used as a reserved character, please follow the next instruction when you want to display those characters. Each character is an ASCII character.

Typing:	Display (in each ""):
\\	"\" itself
\{	"{" itself
\}	"}" itself
\[	"[" itself
\]	"]" itself

## 4. Usual Lyric Display

### 4-1. Japanese Lyric Display

#### 4-1-1. Having one Lyric/Display Meta event responding each one of melody

We recommend having one Lyric/Display Meta event to respond to each one of melody in the same timing with Note On. We also recommend adding Ruby whenever Kanji characters are used.

#### 4-1-2. Ten: Japanese Comma (、), Maru: Japanese Period (。), Space between words...etc

Ten: Japanese Comma (、), Maru: Japanese Period (。), Space between words...etc. should be placed at the end of the latest Lyric/Display Meta Event.

Control Characters (CR、 LF、 HT) should be typed as independent events. When "\" was used

with characters for substitution, the same rule should be applied.

Example)

きらきら ひかる  
ちいさな ほしよ

```
5:4:095<Meta>Lyric/Display "{@JP}" (5)
6:1:000<Meta>Lyric/Display "き" (2)
6:2:000<Meta>Lyric/Display "ら" (2)
6:3:000<Meta>Lyric/Display "き" (2)
6:4:000<Meta>Lyric/Display "ら " (4)
7:1:000<Meta>Lyric/Display "ひ" (2)
7:2:000<Meta>Lyric/Display "か" (2)
7:3:000<Meta>Lyric/Display "る" (2)
7:4:000<Meta>Lyric/Display "\r" (2)
8:1:000<Meta>Lyric/Display "ち" (2)
8:2:000<Meta>Lyric/Display "い" (2)
8:3:000<Meta>Lyric/Display "さ" (2)
8:4:000<Meta>Lyric/Display "な " (4)
9:1:000<Meta>Lyric/Display "ほ" (2)
9:2:000<Meta>Lyric/Display "し" (2)
9:3:000<Meta>Lyric/Display "よ" (2)
9:4:000<Meta>Lyric/Display "\r" (2)
```

## 4-2. European Language (we call it “English” hereafter) Lyric Display

### 4-2-1. One syllable is one Lyric/Display Meta Event

One syllable is described as one Lyric/Display Meta Event. We recommend to have one Lyric /Display Meta event responding each one of melody in the same timing with Note On just as we apply for Japanese Lyric Display.

### 4-2-2. Space comes after each word.

When you type English (or other European) Lyrics, there should be one space between words. This “one space” is the symbol to show a borderline between words. This is a very natural typing manner. If there is no space after Lyric/Display Meta Event, the next Lyric/Display Meta Event will be considered as a part of the latest event. In software for music scores, hyphen (-) is placed when it is necessary. This is used always except when the syllable ends with a space followed. (When a word has more than one syllable, each syllable between the top and the end needs this.)

### 4-2-3. Punctuation marks (comma, period, etc)

Punctuation marks should be placed at the end of a syllable in the event and they should be put before the space. This is also a natural typing manner. Punctuation marks are period (full stop), question mark, etc. In English, the last event of the sentence will be at the last syllable of the last word, then a period (full stop) and a space should follow.

### 4-3. When English Lyrics and Japanese lyrics are mixed in one song

Example)

そうさ baby 今夜は all through the night  
お願い朝まで hold me tight

MS-Kanji (Shift-JIS) character set has ASCII character set (JIS Roman Character set) (Example: A-Z, a-z, 0-9. etc). It is not necessary to type English character code everytime characters switch from Japanese to English. The words like "baby", "all through the night", or "hold me tight" will be displayed normally. However, MS-Kanji (Shift-JIS) code set does not have some special Latin alphabet characters, i.e. accent egule, accent grave or umlaut, etc. Those alphabet parts will not be displayed properly as shown below.

Example: when they are typed)

恋の leçon  
ささやいて Question et réponse

Example: when they are displayed)

恋の 師 oile  
ささやいて Question et r 姿 onse

To solve this problem, type character code information everytime languages change.

Example)

```
09:04:095<Meta>Lyric/Display "{@JP}" (5)
10:01:000<Meta>Lyric/Display "恋[こ" (5)
10:03:000<Meta>Lyric/Display "い]" (3)
10:04:000<Meta>Lyric/Display "の" (2)
11:01:095<Meta>Lyric/Display "{@LATIN}" (8)
11:02:000<Meta>Lyric/Display "le-" (3)
11:01:072<Meta>Lyric/Display "çon" (3)
11:04:000<Meta>Lyric/Display "\r" (2)
12:02:000<Meta>Lyric/Display "{@JP}" (5)
12:02:000<Meta>Lyric/Display "さ" (2)
12:02:024<Meta>Lyric/Display "さ" (2)
12:02:072<Meta>Lyric/Display "や" (2)
12:03:000<Meta>Lyric/Display "い" (2)
12:03:048<Meta>Lyric/Display "て" (2)
12:03:095<Meta>Lyric/Display "{@LATIN}" (8)
12:04:000<Meta>Lyric/Display "Ques-" (5)
13:01:000<Meta>Lyric/Display "tion " (5)
13:02:048<Meta>Lyric/Display "et " (3)
13:02:072<Meta>Lyric/Display "ré-" (3)
13:03:000<Meta>Lyric/Display "ponse" (5)
14:01:000<Meta>Lyric/Display "\r" (2)
14:01:095<Meta>Lyric/Display "{@JP}" (5)
```

Because ANSI character code set does not have any Japanese Kana or Kanji characters, Japanese lyrics will be displayed inappropriately and it will not be readable if you did not type appropriate character code information. Please pay extra attention, especially when you switch code set temporarily in Japanese Lyrics.

#### **4-4. Number of characters for one line**

In some environments, the number of characters for one line (from a carriage return to the next carriage return) is limited. It is safe to have less than 40 characters (1 byte character) including spaces.

#### **Footnote:**

##### 1 MS-Kanji (Shift-JIS) Character Code Set:

MS-Kanji is an abbreviation for “Microsoft Kanji” and another name for “Shift-JIS”. It is called, “SJIS”, too. “JIS” is a code set that is defined by “Japan Industrial Standard”. “Shift-JIS” got its name because 2-byte character code had to shift to avoid 1 byte katakana code area. This is the most popular way to use symbols among Japanese PCs. It contains Hiragana, Katakana, Kanji and Romaji, etc.

##### 2 ANSI Character Code Set:

This code set was designed by ANSI (American National Standards Institute) that regulates standard and does its publicity in the United States. This is a standard that was approved by ISO, too. This code set has 7-bit code set (128 characters) which is called “ASCII character code, and 8 bit expanded characters such as Latin Alphabet (i.e. à, é).

##### 3 ASCII (American Standard Code for Information Interchange) Code:

This is a character set that was approved by ISO 646-1991. It contains 94 printable characters, empty character (21-7F) and 33 control characters (00-20).