

# VLMF Specification

Vodafone Live Melody Format

Version 1.1

10/27/2004

Copyright© 2004 Vodafone Group and FueTrek Co., Ltd.

ALL RIGHTS RESERVED. NO PART OF THIS DOCUMENT MAY BE REPRODUCED OR TRANSMITTED IN ANY FORM OR BY ANY MEANS, ELECTRONIC OR MECHANICAL, INCLUDING INFORMATION STORAGE AND RETRIEVAL SYSTEMS, WITHOUT PERMISSION IN WRITING FROM VODAFONE GROUP AND FUETREK CO., LTD..

**Vodafone Group**

<http://www.vodafone.com/>

**FueTrek Co., Ltd.**

<http://www.fuetrek.co.jp/>

## Revision History

Version	Date	Description
1.00	10/25/2004	The first release.
1.10	10/27/2004	Format name change

## Trademarks

GIF is a service mark of CompuServe Incorporated.

# Contents

1.	Introduction and Background.....	4
2.	Definitions of Terms .....	5
3.	Overview of VLMF .....	6
3.1.	Extension to Mobile XMF .....	6
3.2.	File Structure of VLMF.....	6
3.3.	Sequence of Playback .....	7
3.3.1.	Event Message in Sequence of Extension.....	7
3.4.	File Layout of VLMF .....	7
3.5.	Guide Line .....	8
4.	Contents Information about VLMF .....	9
4.1.	Required Information.....	9
4.1.1.	Title .....	9
4.1.2.	Version of VLMF .....	9
4.2.	Optional Information.....	9
4.2.1.	Vendor Name .....	9
4.2.2.	Copyright.....	9
4.2.3.	Copyright Code.....	9
4.2.4.	Version of Contents.....	9
5.	Resource Format ID and Assignment.....	10
6.	Sync Extension File Node.....	11
6.1.	Node Header.....	11
6.2.	Sync Extension File .....	11
6.2.1.	File Header .....	11
6.2.2.	Setting Data.....	11
7.	Wave Audio File Node .....	13
7.1.	Wave Audio Group.....	13
7.2.	Node Header.....	14
7.3.	Wave Audio File .....	14
7.3.1.	File Header .....	14
7.3.2.	Wave Data Chunks.....	14
7.3.3.	Sequence Data Chunk.....	15
8.	Simple Image File Node.....	17
8.1.	Node Header.....	17
8.2.	Simple Image File.....	17
8.2.1.	File Header .....	17
8.2.2.	Image Data Chunk .....	18
8.2.3.	Sequence Data Chunk.....	18
9.	Reference .....	20

# 1. Introduction and Background

This document specifies the music format called VLMF extended for more sophisticated Vodafone mobile phone functionality, and it is based on Mobile XMF.

Recently, the performance of mobile phone is greatly increasing and it is also required to play music with high quality and advanced expression. But current Mobile XMF specification that is developed as a standard for mobile phone can only treat SMF (SP-MIDI) and Mobile DLS. Moreover, it is not considered about synchronization with Wave audio, LED and vibration alarm, although such function has already been implemented by the mobile phone on the market.

VLMF specifies the data format to implement these functions and to synchronize with MIDI based on Mobile XMF. Any change need not be added to the existing parts by the enhancement of specification. It is the same also about an authoring tool, and it can use the existing authoring tool except an extended portion.

The example of application that VLMF assumes is shown below.

- The rich ringtone using DLS and Wave sound
- The flashing LED and vibration alarm synchronized with ringtone
- Greeting mail with ringtone and images

Regarding technical content/term/wording/figure/etc appears in this document and/or corresponding specifications/references may not be suitable for actual implementation due to User Interface consistency or legal protection. (E.g. In case of using/infringing brand name, portrait, and intellectual property rights of any third party and etc)  
Manufactures must confirm and use suitable ones for actual implementation.

Regarding file operations of Copyright protected file. Refer Content Types Terminal Capability Definition and handle according to definitions in the Terminal Capability Definition.

VLMF is formulated by Vodafone Group and FueTrek Co., Ltd. Collaboratively.

## 2. Definitions of Terms

- **Mobile XMF**  
Specification of the method for bundling Standard MIDI Files with Mobile DLS instrument data. See definition in [4].
- **XMF**  
Acronym for eXtensible Music Format as defined in [1] and modified in [2],[3].
- **Mobile DLS**  
Specification for waveform-based musical instrument content format. See definition in [7].
- **SMF**  
Acronym for Standard MIDI File (see below). See definition in [5].
- **SP-MIDI**  
Acronym for Scalable Polyphony MIDI. See definition in [6]
- **Standard MIDI File**  
Specification for MIDI message content format. See definition in [5].
- **VLQ**  
Variable Length Quantity, a binary number format with a variable number of bytes. See definition in Section 4.1 of [1].
- **XString**  
eXtensible String, a text string format with a leading length integer in VLQ form and no terminating character. See definition in section 4.2 of [1].
- **player**  
Program or device for playing VLMF files.

### 3. Overview of VLMF

#### 3.1. Extension to Mobile XMF

Additional functions of VLMF that differs from Mobile XMF list below. The specifications except these are the same as those of mobile XMF.

- Contents Information about VLMF

It specifies contents management information. (Refer section 4)

- SMF Extension

It specifies the setting of LED and Vibrator to synchronize with MIDI playback. (Refer section 6)

- Wave Audio

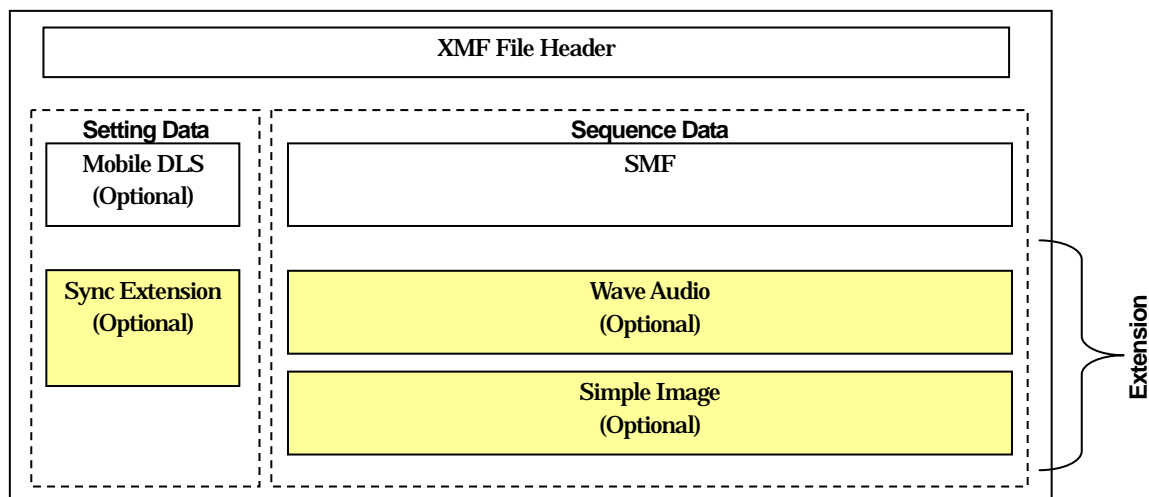
It specifies the Wave Audio data and the sequence to be played in playing MIDI. (Refer section 7)

- Simple Image

It specifies the image data and the sequence to be displayed in playing MIDI. (Refer section 8)

#### 3.2. File Structure of VLMF

The following figure shows the file structure and logical hierarchy of VLMF.



**Figure 3-1 File structure of VLMF**

\*"Setting Data" means the data that is loaded before a playback and does not contain a playback sequence.

\*"Sequence Data" means the data containing the playback sequence itself.

\*SMF and Mobile DLS (Optional) have already been defined by Mobile XMF.

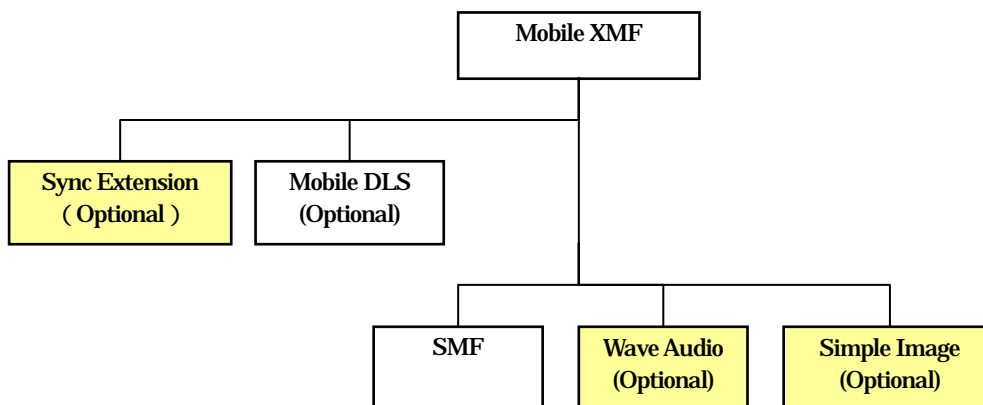


Figure 3-2 Logical hierarchy of VLMF

### 3.3. Sequence of Playback

The following figure shows the concept of the playback sequence by VLMF. As shown in a figure, each sequence of MIDI, Wave Audio and Simple Image starts to play simultaneously. When all sequence is completed, the whole sequence finishes playback.

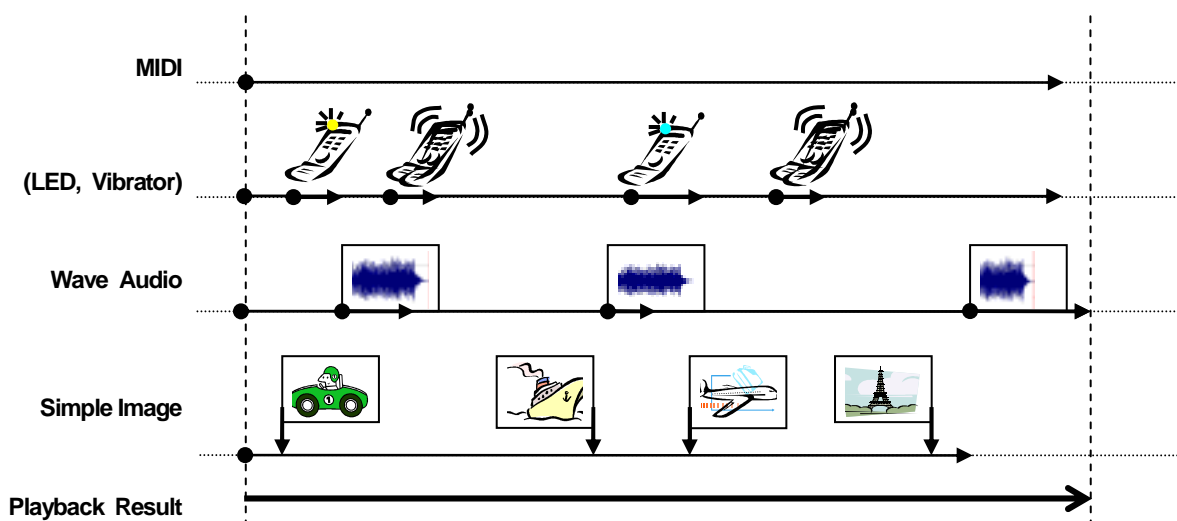


Figure 3-3 Sequence of playback by VLMF

#### 3.3.1. Event Message in Sequence of Extension

The event message used in additional sequence of VLMF consists of the set of Delta Time and Event Message as with SMF format.

##### 3.3.1.1. Delta Time

Delta Time designates the time to make event in variable length (VLQ form).

**\*Note:** The time unit of the delta time used in all sequence depends on the Time Base and the Tempo that are specified in SMF.

##### 3.3.1.2. Event Message

Each event message basically treats as original specification.

### 3.4. File Layout of VLMF

VLMF has one or zero file node of SMF Extension, Wave Audio and Simple Image respectively. They are optional and they belong to RootNode as well as Mobile DLS File and SMF File. The following table shows the example file layout of In-Line Resource case.

FileHeader		
Tree	RootNode	Mobile DLS FileNode (Optional)
		SMF FileNode
		Sync Extension FileNode (Optional)
		Wave Audio FileNode (Optional)
		Simple Image FileNode (Optional)

**Figure 3-4 File Layout of VLMF**

**\*Note:** Each file of extended specification defined in VLMF must appear after the SMF file. The order of them isn't required.

### 3.5. Guide Line

Recommended behavior of the player that does not support VLMF is specified as follows.

- Recognize the SMF file and the Mobile DLS file correctly.
- Ignore Meta-Data defined as VLMF.
- Ignore data file defined as VLMF.

Besides, the player which supports VLMF must correctly recognize not only the data defined by this document but also Mobile XMF.



## 4. Contents Information about VLMF

The information about VLMF is defined as Meta-Data of Root Node. The information of RootNode has priority over the information in the other node. The order of the Meta-Data shown below isn't important.

### 4.1. Required Information

The following information must specify.

#### 4.1.1. Title

It specifies title of data. It uses the Standard Field as the Meta-data field and the FieldID is 8.

It uses the Universal or International Contents as the FieldContents. In using the International Contents, it specifies the Country/Language Code by the Country/Language Code Table that is defined in the FileHeader.

#### 4.1.2. Version of VLMF

It specifies a version of VLMF. It uses the Custom Field as the Meta-data field. And the Field name designates "Version of VLMF" (Xstrings from). It uses the Universal Contents as the FieldContents and the StringFormatTypeID is 6. The size of FieldContents is 4 bytes. It specifies 0x0100 as VLMF Version 1.0.

### 4.2. Optional Information

The following information may not specify.

#### 4.2.1. Vendor Name

It specifies a vender name. It uses the Custom Field as the Meta-data field. And the Field name designates "Vender Name" (Xstrings from). It uses the International Contents as the FieldContents and it specifies the Country/Language Code by the Country/Language Code Table that is defined in the FileHeader.

#### 4.2.2. Copyright

It specifies copyright information. The way of definition of data is similar to section 4.2.1, but the Field name designates "Copyright".

#### 4.2.3. Copyright Code

It specifies the code for copyright management. It uses the Custom Field as the Meta-data field. And the Field name designates "Copyright Code" (Xstrings from). It uses the Universal Contents as the FieldContents and the StringFormatTypeID is 0. The FieldContents is specified in ASCII text. The code is separately defined in an operation phase.

#### 4.2.4. Version of Contents

It specifies a version of contents. It uses the Custom Field as the Meta-data field. And the Field name designates "Version of Contents" (Xstrings from). It uses the Universal Contents as the FieldContents and the StringFormatTypeID is 6. The size of FieldContents is 4 bytes.

## 5. Resource Format ID and Assignment

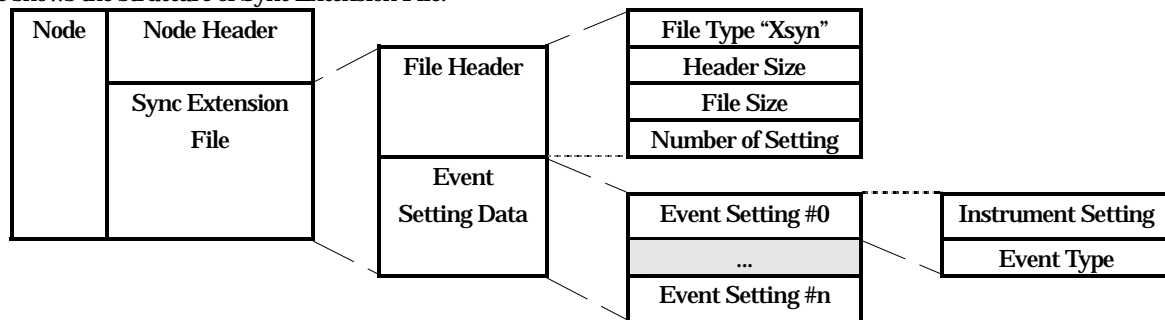
Each data defined as VLMF uses 1 as FormatTypeID, and uses "00H 40H 01H" as MMA Mfr. ID of Vodafone K.K, and uses the following definitions as ResourceFormatID.

**Table 5-1 Resource Format ID of VLMF**

ResourceFormatID	Format Name
0	Sync Extension
1	Wave Audio
2	Simple Image

## 6. Sync Extension File Node

Sync Extension defines the setting to synchronize LEDs and vibrator with Note On and Note Off of specified instrument setting. It can have 32 settings in maximum. The operation of LEDs and vibrator determines by combining all settings. The following figure shows the structure of Sync Extension File.



**Figure 6-1 Structure of Sync Extension file node**

### 6.1. Node Header

Resource Format ID designates the data type in Meta Data Field. (Refer section 5 about Resource Format ID and its assignment.)

### 6.2. Sync Extension File

#### 6.2.1. File Header

##### 6.2.1.1. File Type

The number of bytes of the File Type information is 4 and the content is "Xsyn".

##### 6.2.1.2. Header Size

Header Size designates the size of the File Header in variable length (VLQ form).

##### 6.2.1.3. File Size

File Size designates the size of File volume in variable length (VLQ form).

##### 6.2.1.4. Number of Setting

This chunk designates the number of event setting in a byte.

### 6.2.2. Setting Data

Event Setting Data consists of sets of 2 event settings that described in following.

#### 6.2.2.1. Instrument Setting

The number of byte of the instrument number to make an event is 4. They consist of Bank (2bytes), Program (1byte) and Key (1byte). Note the setting of Key is available only for a drum instrument.

### 6.2.2.2. Event Type

Event Type designates event type in each bit of a byte. It can select LEDs and vibrator. They light up or start vibrating at the timing of Note On, and they also turn off a light or stop vibrating at the timing of Note Off. There are three settings of R, G and B for the LEDs, and they correspond to Red, Green, and Blue respectively. If the device doesn't support color lighting, specifying any of R, G and B means lighting LED.

**Table 6-1 Event Type**

Bit	Event Type
0	LED1 (Red LED for color device, any LED for non-color device)
1	LED2 (Green LED for color device, any LED for non-color device)
2	LED3 (Blue LED for color device, any LED for non-color device)
3	Vibrator
Others	Reserved

## 7. Wave Audio File Node

Wave Audio allows playing the specified Wave Data according to an attached sequence in playing MIDI. Thereby, it is possible to play Wave Data synchronizing with playback of MIDI. It can be included one or more Wave Data in the file and each format need not be the same. It uses general standard Wave formats. (See the Table 7-1)

There is only one track for Wave Audio, and it can define maximally 16 Wave Groups. Each Wave Data specifies the Wave Group that plays when it starts to note on. The following figure shows the structure of Wave Audio File.

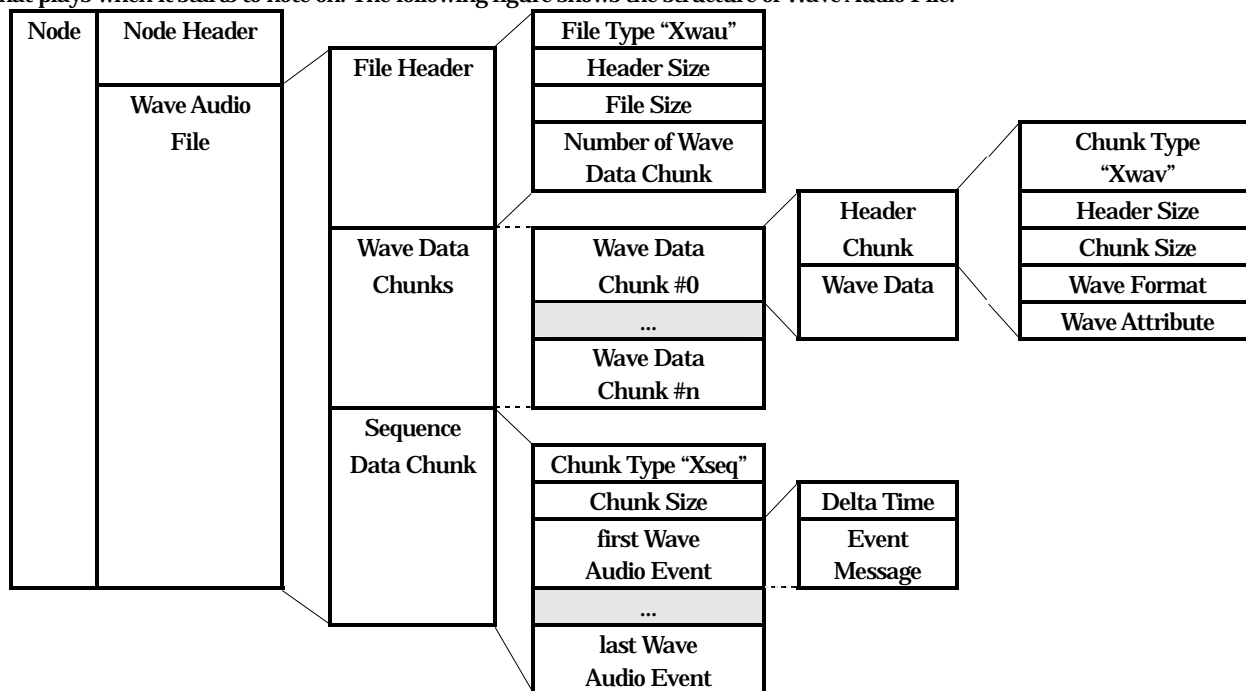


Figure 7-1 Structure of Wave Audio file node

### 7.1. Wave Audio Group

Wave Audio Group is a unit for managing the setting of Volume, Pan and others. By using this, it becomes easy to create the sequence data because it doesn't need to set the Volume and Pan every time Wave Audio Data sets Note On. It is possible to contain multiple Note On to play simultaneously in one Wave Audio Group, but the number of Wave Data possible to play simultaneously depends on actual implementation. If it does Note On or Preload exceeding the maximum number of Wave Data possible to play simultaneously on an actual device, it stops a Wave Data, which started in the oldest time and plays new Wave Data.

**\*Note:** The device must have one resource to play wave audio at least.

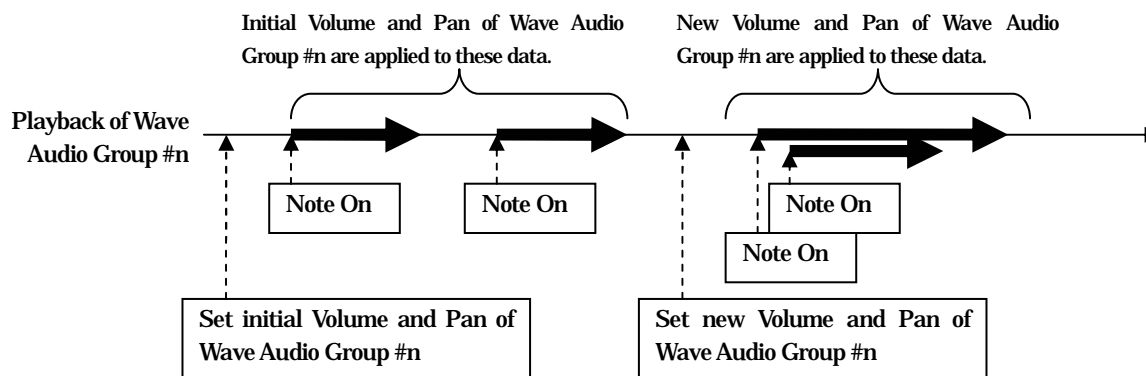


Figure 7-2 Wave Audio Group

## 7.2. Node Header

Resource Format ID designates the data type in Meta Data Field. (Refer section 5 about Resource Format ID and its assignment.)

## 7.3. Wave Audio File

Sequence Data Chunk should be located at the end of the file.

### 7.3.1. File Header

#### 7.3.1.1. File Type

The number of bytes of the File Type information is 4 and the content is "Xwau".

#### 7.3.1.2. Header Size

Header Size designates the size of the File Header in variable length (VLQ form).

#### 7.3.1.3. File Size

File Size designates the size of File volume in variable length (VLQ form).

#### 7.3.1.4. Number of Wave Data

This Chunk designates the number of Wave Data Chunk in a byte.

### 7.3.2. Wave Data Chunks

It contains one or more Wave Data Chunk. The ID is assigned sequential number start with 0.

#### 7.3.2.1. Header Chunk

**\*Note:** If the number of Chunk is less than the specified number in File Header, all Wave Data Files are assumed as invalid and should be ignored. If the number of Chunks is greater than the specified number in File Header, data is valid for the number designated.

##### 7.3.2.1.1. Chunk Type

The number of bytes of the Chunk Type information is 4 and the content is "Xwav".

##### 7.3.2.1.2. Header Size

Header Size designates the size of the Chunk Header in variable length (VLQ form).

##### 7.3.2.1.3. Chunk Size

Chunk Size designates the size of Chunk volume in variable length (VLQ form).

##### 7.3.2.1.4. Wave Format

Wave Format is designated in string (Xstring form). The string uses "Wave Format String" of the following table. The formats to be supported is Linear PCM, and support of G.726 ADPCM is strongly desirable (support of any other formats are optional). Following table show the sampling rate and bit rate to be supported in each format.

**Table 7-1 Wave format to be supported**

Format	Wave Format String	Specification	WFormatTag	Channel	Sampling rate (Hz)	Bit rate (bit)
audio/x-wav	PCM	Microsoft Linear PCM	0x000	Monaural / Stereo	8K / 16K	8, 16
				Monaural	32K	
audio/x-wav	G726_ADPCM	G.726 ADPCM	0x0064	Monaural / Stereo	8K / 16K	4
				Monaural	32K	

**\*Note:** It assumes Stereo/Mono, Sampling Frequency, and Bit/Sample is defined in Wave Data.

Support of G.726 ADPCM is not mandatory, but strongly desirable.

##### 7.3.2.1.5. Wave Attribute

Wave Attribute designates the playback attribute in a byte. This can select whether to play Wave Audio Data when the player

plays in new key or tempo that is different from original ones. For instance, the vocal that is created with Wave Audio Data might be off key when the player plays in different tempo. It can be avoided or be permitted in this setting.

0: It is played always.

1: It is not played when Key or Tempo is changed.

others: Reserved

### 7.3.2.2. Wave Data

It contains Wave Data. The data format follows the specification of the WAVE format that is sub-set of Microsoft's RIFF specification, and the wave data is stored from "fint" sub-chunk.

## 7.3.3. Sequence Data Chunk

Sequence Data Chunk contains the list of the time information and events controlling Wave Audio.

### 7.3.3.1. Chunk Type

The number of bytes of the Chunk Type information is 4 and the content is "Xseq".

### 7.3.3.2. Chunk Size

Chunk Size designates the size of the Sequence Data in variable length (VLQ form).

### 7.3.3.3. Wave Audio Event

Wave Audio Event consists of the time information (Delta Time) and the defined events in the following table. Refer section 3.3.1.1 for the information of Delta Time.

**Table 7-2 Event messages of Wave Audio**

Message	Event name
0x8n	Note Off
0x9n	Note On
0xA $n$	Preload
0xB $n$	Group Volume / Group Pan
0xD0	Wave Audio Master Volume
0xFF	End of Track

**\*Note:**  $n$  in the table represents targeted Wave Audio Group.

#### 7.3.3.3.1. Note Off

The Note Off event stops playing for specified Wave Audio Group and Wave Data ID.

Byte	Value	Description
0	0x8 $n$	Note Off message ID (8) / Wave Audio Group ( $n=0\sim15$ )
1	$n$	Wave Data ID (0~255)

**\*Note:** It must ignore the event in the following cases.

- Specified Wave Data ID is invalid.
- Player cannot identify Wave format of specified Wave Data ID.
- Playback of specified Wave Data ID has already been finished.

#### 7.3.3.3.2. Note On

The Note On event starts playing for specified Wave Audio Group with Wave Data ID and Velocity. If there is Preload event with the same setting, it uses the Preload Data. The Wave Audio data which starts playing by Note On event finishes playing at the end of data automatically.

Byte	Value	Description
0	0x9 $n$	Note On message ID (9) / Wave Audio Group ( $n=0\sim15$ )
1	$n$	Wave Data ID (0~255)
2	$n$	Velocity (0~127)

**\*Note:** The base unit and variant curve of Velocity depend on an actual implementation.

**\*Note:** It must ignore the event in the following cases.

- Specified Wave Data ID is invalid.
- Player cannot identify Wave format of specified Wave Data ID.
- It specifies Velocity from 128 to 255.

## 7.3.3.3.3. Preload

The Preload event preloads required Wave Data specified by Wave Data ID and Wave Audio Group. When player need to preload prior to Note On, the event is used. If preload is not needed, the event is not used..

Byte	Value	Description
0	0xA $n$	Preload message ID (A) / Wave Audio Group ( $n=0\sim15$ )
1	$n$	Wave Data ID (0~255)

**\*Note:** It must ignore the event in the following cases.

- Specified Wave Data ID is invalid.
- Player cannot identify Wave format of specified Wave Data ID.

**\*Note:** The data which is called by Preload event must be called by Note On event.

## 7.3.3.3.4. Group Volume

The Group Volume event sets specified Volume for specified Wave Audio Group.

Byte	Value	Description
0	0xB $n$	Control message ID (B) / Wave Audio Group ( $n=0\sim15$ )
1	0x07	Control number
2	$n$	Volume (0~127)

**\*Note:** The base unit and variant curve of Volume depend on an actual implementation.

**\*Note:** If it specifies Volume from 128 to 255, it must ignore the event.

## 7.3.3.3.5. Group Pan

The Group Pan event sets specified Panpot for specified Audio Group. Setting of 0 is Left-End, 64 is Center and 127 is Right-End.

Byte	Value	Description
0	0xB $n$	Control message ID (B) / Wave Audio Group ( $n=0\sim15$ )
1	0x0A	Control number
2	$n$	Pan (0~127)

**\*Note:** The base unit of Panpot depends on an actual implementation.

**\*Note:** If it specifies Panpot from 128 to 255, it must ignore the event.

## 7.3.3.3.6. Wave Audio Master Volume

The Wave Audio Master Volume event sets Master Volume for all Wave Audio. This setting influences all Wave Audio Groups.

Byte	Value	Description
0	0xD0	Control message ID (D0)
1	$n$	Volume (0~127)

**\*Note:** The base unit and variant curve of Volume depend on an actual implementation.

**\*Note:** If it specifies Volume from 128 to 255, it must ignore the event.

## 7.3.3.3.7. End of Track

End of Track shows the end of sequence. This is the same as specification of Meta Event of SMF. The events that appear after this event are ignored.

Byte	Value	Description
0	0xFF	Meta Event message ID
1	0x2F	Command ID
2	0x00	-

**\*Note:** It must insert in the end of sequence data.



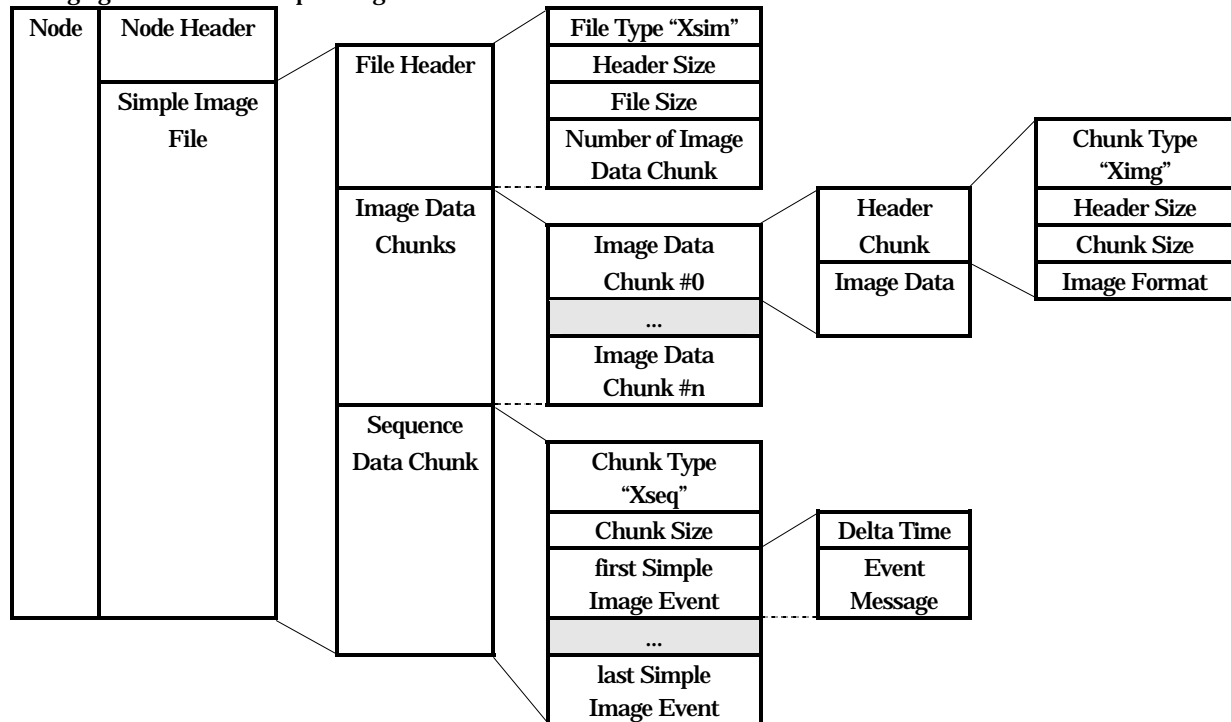
## 8. Simple Image File Node

Simple Image data defines image information, which is shown synchronized with MIDI or Wave Audio data.

The typical application is Greeting Card message, which flips piece of images while playing the MIDI music. The Sequence data is used to synchronize the Image and MIDI or Wave Audio.

Multiple Image Data can be used and the format can be different. A general image format is used for Image Data.(See Table 8-1)

Following figure shows the Simple Image File Node structure.



**Figure 8-1 Structure of Simple Image file node**

**\*Note:** The size of image data must be same or smaller than the size of display area of device. The device draws the image data in the center of display area. If the size of image data is larger than the size of display area, the behavior depends on the device. (It may display center area of image data, or display it after scaling down.)

**\*Note:** If the size of display device is QVGA, the size of display area of device must have 240pixels width and 260pixels height. If it is the other size, the size of display area of device depends on the implementation.

### 8.1. Node Header

Resource Format ID designates the data type in Meta Data Field. (Refer section 5 about Resource Format ID and its assignment.)

### 8.2. Simple Image File

Sequence Data should be located at the end of File.

#### 8.2.1. File Header

##### 8.2.1.1. File Type

The number of bytes of the File Type information is 4 and the content is "Xsim".

##### 8.2.1.2. Header Size

Header Size designates the size of the File Header in variable length (VLQ form).

##### 8.2.1.3. File Size

File Size designates the size of File volume in variable length (VLQ form).

#### 8.2.1.4. Number of Image Data Chunk

This Chunk designates the number of Image Data Chunk in a byte.

#### 8.2.2. Image Data Chunk

One or multiple numbers of Image Data File are stored in the Image Data Chunk. The ID is assigned sequential number start with 0.

##### 8.2.2.1. Header Chunk

**\*Note:** When the number of Chunk is less than the value in File Header Chunk, Simple Image File is invalid and should be ignored. When the number of Chunk is greater than the in File Header Chunk, data is valid for the number designated.

##### 8.2.2.1.1. Chunk Type

The number of bytes of the Chunk Type information is 4 and the content is "Ximg".

##### 8.2.2.1.2. Header Size

Header Size designates the size of the Chunk Header in variable length (VLQ form).

##### 8.2.2.1.3. Chunk Size

Chunk Size designates the size of the Image Data Chunk in variable length (VLQ form).

##### 8.2.2.1.4. Image Format

Image Format is designated in string (Xstring form). The string uses "Image Format String" of the following table. The formats to be supported are GIF87a, GIF89a and PNG.

**Table 8-1 Image format to be supported**

Format	Image Format String	Specification	Description
image/gif	GIF87a	GIF87a	Animate GIF doesn't need to support
Image/gif	GIF89a	GIF89a	Animate GIF doesn't need to support
image/png	PNG	PNG 1.0	Only 256 index color mode needs to support

**\*Note:** Size of image and color mode should be defined in Image Data.

##### 8.2.2.2. Image Data

It contains Image Data The data format follows specified specification, and the Image Data is stored whole data of its specification.

#### 8.2.3. Sequence Data Chunk

The Sequence Data Chunk is the correction of set of time and Image control events.

##### 8.2.3.1. Chunk Type

The number of bytes of the Chunk Type information is 4 and the content is "Xseq".

##### 8.2.3.2. Chunk Size

Chunk Size designates the size of the Sequence Data in variable length (VLQ form).

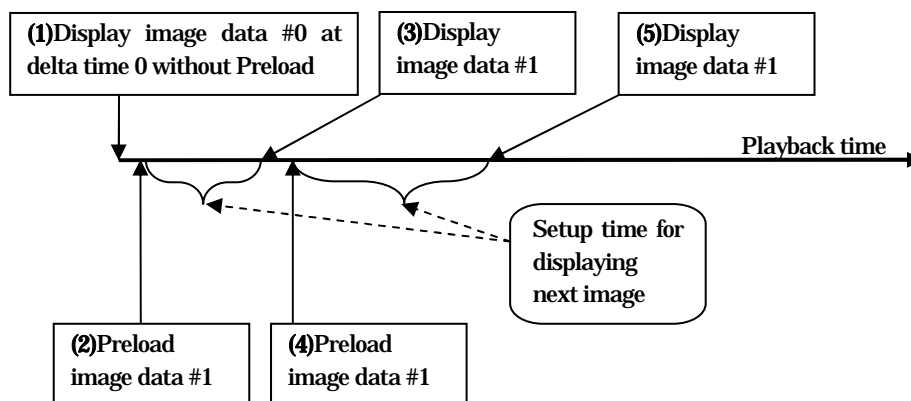
##### 8.2.3.3. Simple Image Event

Simple Image Event consists of time information (Delta Time) and one of events defined in table 10-2. Refer section 3.3.1.1 for the Delta Time explanation.

**Table 8-2 Event messages of Simple Image**

Message	Event name
0x90	Display
0xA0	Preload
0xFF	End of Track

Figure 8-2 shows the example of Simple Image sequence using each Event message of Preload and Display.



**Figure 8-2 Example of Simple Image sequence**

#### 8.2.3.3.1. Display

Display Event designates the ID of Image Data to cause events.

Byte	Value	Description
0	0x90	Display message ID (90)
1	<i>n</i>	Image Data ID (0~255)

**\*Note:** Events are ignored in following cases.

- The Image Data ID is invalid
- Player could not recognize the Image Data for ID

#### 8.2.3.3.2. Preload

The Preload event preloads required Image Data specified by Image Data ID. When player need to preload prior to Display Event, the event is used. If preload is not needed, the event is not used.

Byte	Value	Description
0	0xA0	Preload message ID (A0)
1	<i>n</i>	Image Data ID (0~255)

**\*Note:** It must ignore the event in the following cases.

- Specified Image Data ID is invalid.
- Player cannot identify Image format of specified Image Data ID.

**\*Note:** The data which is called by Preload event must be called by Display event.

#### 8.2.3.3.3. End of Track

Refer section 7.3.3.3.7.

## 9. Reference

- [1] "Specification for XMF Meta File Format", RP-030, MIDI Manufacturers Association, Los Angeles, CA, USA, 2001
- [2] "XMF Meta File Format Updates 1.01", RP-039, MIDI Manufacturers Association, Los Angeles, CA, USA, 2003
- [3] "XMF Meta File Format 2.00", RP-0xx, MIDI Manufacturers Association, Los Angeles, CA, USA, 2004
- [4] Mobile XMF Content Format Specification, RP-0xx, MIDI Manufacturers Association, Los Angeles, CA, USA. 2004.
- [5] "Standard MIDI Files", RP-001, in the Complete MIDI 1.0 Detailed Specification, Document Version 96.1, MIDI Manufacturers Association, Los Angeles, CA, USA, 1996
- [6] "Scalable Polyphony MIDI Specification, Version 1.0", RP-034, MIDI Manufacturers Association, Los Angeles, CA, USA, February 2002
- [7] "Mobile DLS Specification", RP-0xx, MIDI Manufacturers Association, Los Angeles, CA, USA, 2003
- [8] "40, 32, 24, 16 kbit/s adaptive differential pulse code modulation (ADPCM)", Recommendation G.726 (12/90), ITU-T, 1990-12
- [9] "PNG (Portable Network Graphics) Specification Version 1.0", RFC-2083, Boutell, T., et. AL, March 1997