

# MA-3 Sound Middleware Specification For SMAF/Phrase MA-2 Compatible API

Version 1.6.0

January 16, 2003

Yamaha Corporation

[Notes]

This document is the specification of MA-3 Sound Middleware as sample source code.  
This explains the expected operation of Sound Middleware, but doesn't guarantee operation of sample middleware.

Copyright to this document is the property of Yamaha Corporation.  
Transfer or copying of this document in part or in whole requires the permission of Yamaha Corporation.  
The contents of this document are subject to change without notice.



Copyright © 2001-2003 YAMAHA Corporation

All rights reserved

## Note:

For explanation of definition of functions, the following formats are defined for use.

Definition format	Meaning	Definition format	Meaning
UINT8	8 bits without code	SINT8	8 bits with code
UINT16	16 bits without code	SINT16	16 bits with code
UINT32	32 bits without code	SINT32	32 bits with code

## Revision

Version	Date	Description
1.0.0	2001.1.10	Initial edition
1.1.0	2001.1.13	Terminate/SetPanpot/GetPanpot was added.
1.2.0	2001.1.31	Pause/Restart was added. Specification of GetStatus was changed.
1.2.1	2001.1.31	Incorrect description was corrected. Descriptions of functions were added.
1.2.2	2001.8.3	Maximum repeated reproduction count was changed from 127 to 255.
1.2.3	2001.8.21	The return value at the error of CheckData() and SetData() were changed. The return value mistake of SetVolume() and SetPanpot() were corrected.
1.3.0	2001.9.4	GetPosition() was added. Seek() was added. The condition at the synchronous setup was added.
1.3.1	2001.9.6	The operating-condition at the synchronous setup was added. The play position at the pronunciation closure was changed. The play position at the Play was specified.
1.3.2	2001.9.12	Play start position setting operation. was changes
1.3.3	2001.10.1	The error condition of SetLink was changed.
1.3.4	2001.11.9	The position after pronunciation closure was corrected. Clerical error were corrected.
1.3.5	2001.11.14	The volume and panpot value setting position were specified. The play position mistake after closure was corrected.
1.4.0	2001.11.15	Chapter 1 was changed. The explanation of SetVolume() and SetPanpot() were added.
1.5.0	2001.11.26	Argument check definition of SetData() was changed. Explanation clerical error of Kill() was corrected.
1.5.1	2001.12.4	Clerical error of Chapter 2 API list were corrected. Explanation of RemoveData() was changed. Explanation of GetLink() was changed.
1.5.2	2002.1.28	Terminate was added to Chapter 3 state transition diagram. Explanation of SetPanpot() was added.
1.6.0	2003.1.16	Chapter 2 Phrase_GetLength was added. Chapter 4.7 The operation specification of Seek was added. Chapter 4.22 Phrase_GetLength was added.

## Contents

<b>1</b>	<b>Introduction.....</b>	<b>4</b>
<b>2</b>	<b>API list .....</b>	<b>4</b>
<b>3</b>	<b>Transition of status .....</b>	<b>5</b>
<b>4</b>	<b>Definition of functions .....</b>	<b>6</b>
4.1	Phrase_Initialize .....	6
4.2	Phrase_Terminate.....	6
4.3	Phrase_GetInfo.....	7
4.4	Phrase_CheckData .....	8
4.5	Phrase_SetData .....	8
4.6	Phrase_GetPosition .....	9
4.7	Phrase_Seek .....	9
4.8	Phrase_Play.....	10
4.9	Phrase_Stop.....	10
4.10	Phrase_Pause.....	10
4.11	Phrase_Restart.....	10
4.12	Phrase_Kill.....	11
4.13	Phrase_SetVolume .....	11
4.14	Phrase_GetVolume .....	11
4.15	Phrase_SetPanpot.....	12
4.16	Phrase_GetPanpot.....	12
4.17	Phrase_GetStatus .....	12
4.18	Phrase_RemoveData .....	13
4.19	Phrase_SetEvHandler.....	13
4.20	Phrase_SetLink .....	14
4.21	Phrase_GetLink.....	14
4.22	Phrase_GetLength.....	14

## 1 Introduction

This document defines MA-2 compatible API (Application Program Interface) when SMAF/Phrase is designated, and present specifications of functions of Sound Middleware that performs the installation. This API shall be used also in MA-3 sound middleware. SMAF/Phrase cannot be controlled by MaSound API.

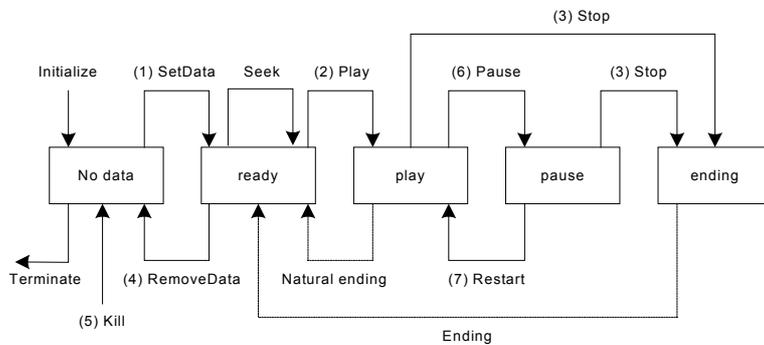
## 2 API list

This chapter shows definitions for SMAF/Phrase reproduction MA-2 compatible API. The group of functions that are covered here are as follows.

Function name	Description
Phrase_Initialize	Initialization of SMAF/Phrase compatible I/F
Phrase_Terminate	Ending of SMAF/Phrase compatible I/F
Phrase_GetInfo	Acquisition of synthesizer information
Phrase_CheckData	Checking of SMAF/Phrase data
Phrase_SetData	Registration of SMAF/Phrase data that is to be played
Phrase_GetPosition	Acquisition of play position
Phrase_Seek	Setting play position
Phrase_Play	Starting play
Phrase_Stop	Stopping play
Phrase_Pause	Pausing play
Phrase_Restart	Cancellation of pausing play
Phrase_Kill	Forced ending
Phrase_SetVolume	Setting volume
Phrase_GetVolume	Acquisition of volume
Phrase_SetPanpot	Setting panpot
Phrase_GetPanpot	Acquisition of panpot
Phrase_GetStatus	Acquisition of status
Phrase_RemoveData	Registration deletion of SMAF/Phrase data
Phrase_SetEvHandler	Setting call back
Phrase_SetLink	Setting of synchronous reproduction sequence
Phrase_GetLink	Acquisition of synchronous setting sequence
Phrase_GetLength	Acquisition of playback time.

### 3 Transition of status

The compatible API takes the following five statuses.



no data : The synthesizer system is in SMAF/Phrase but is not used.

ready : Data are loaded and sequencer is initialized.

play : Reproduction is in progress.

pause : Pausing is in progress.

ending : Waiting for ending of ending processing

#### (1) SetData

SMAF/Phrase is loaded with `Phrase_SetData ()`, and tone generation mechanism is initialized.

#### (2) Play

Play is started with `Phrase_Play ()`.

#### (3) Stop

Play is stopped with `Phrase_Stop ()`.

#### (4) RemoveData

Sequence data is cleared and setting of tone generation mechanism is cleared with `Phrase_RemoveData ()`.

#### (5) Kill

Phrase sequencer unused state is restored with `Phrase_Kill ()`.

#### (6) Pause

Play is paused with `Phrase_Pause ()`.

#### (7) Restart

Pausing of play is cancelled with `Phrase_Restart ()`.

## 4 Definition of functions

### 4.1 Phrase\_Initialize

int Phrase\_Initialize (void)

---

Description

Initializes SMAF/Phrase compatible I/F  
Tone generation resources that are needed for reproduction of SMAF/Phrase are secured.  
Sets playback volume and panpot value as default value.

Argument

None

Returned value

0                    Normal ending  
-1                    Error

### 4.2 Phrase\_Terminate

int Phrase\_Terminate (void)

---

Description

Performs ending processing of SMAF/Phrase compatible.  
Tone generation resources that are needed for reproduction of SMAF/Phrase are released.

Argument

None

Returned value

0                    Normal ending  
-1                    Error

### 4.3 Phrase\_GetInfo

int Phrase\_GetInfo (struct info \*dat)

Description

Acquires synthesizer information.

Argument

dat

Pointer for synthesizer information structural body

struct info{

long MakerID;//Manufacturer's ID (Conforms to MIDI-SysEx)

1 byte ID : 0x00010000 - 0x007F0000

3 byte ID : 0x00000001 - 0x00007F7F

Out of MIDI standard : 0x00800000 - 0x00FFFFFF

(Managed by SMAF manager)

int DeviceID;//DeviceID (Conforms to MIDI-SysEx)

int VersionID;//version No. (optional)

int MaxVoice;// Max. no. of generated tones per one SMAF/Phrase

int MaxChannel;//No. of SMAF/Phrases that can be reproduced simultaneously

int SupportSMAF;// SMAF type that can be supported

bit0 : Compatible with SMAF/Phrase Ver1.0

long Latency;//Synthesizer response time (us)

-1 means unknown.

}

Set value for MA-3

MakerID = 0x00430000

DeviceID = 0x06

VersionID = 1

MaxVoice = 4

MaxChannel = 4

SupportSMAF = 1

Latency = 20000

Returned value

0

Normal ending

-1

Error

#### 4.4 Phrase\_CheckData

int Phrase\_CheckData (unsigned char \*data,unsigned long len)

Description

Check format of SMAF/Phrase

Argument

data                    Pointer for sequence data  
len                      Length of sequence data [bytes]

Returned value

0                        Normal ending  
-1                       Error

#### 4.5 Phrase\_SetData

int Phrase\_SetData (int ch,unsigned char \*data,unsigned long len,int check)

Description

Designates SMAF/Phrase that is to be reproduced. (including format check)  
The playback start position after SetData becomes data head.

Argument

ch                       Sequence No (0 to 3)  
data                     Pointer for Sequence data  
len                      Length of sequence data[bytes]  
check                    Check designation  
                          0 : No error checking  
                          != 0 : With error checking

Returned value

0                        Normal ending  
-1                       Error



#### 4.8 Phrase\_Play

int Phrase\_Play (int ch,int loop)

---

Description

The play start from play start position. Issue to slave assignment data is taken as an error.

Argument

ch	Sequence No (0 to 3)
loop	Loop count (0..255) 0 : endless repeat, 1 : one shot

Returned value

0	Normal ending
-1	Error

#### 4.9 Phrase\_Stop

int Phrase\_Stop (int ch)

---

Description

Stops the play. Issue to slave assignment data is taken as an error.

The play position after Stop becomes the head. It is the same in the case of natural ending.

Argument

ch	Sequence No (0 to 3)
----	----------------------

Returned value

0	Normal ending
- 1	Error

#### 4.10 Phrase\_Pause

int Phrase\_Pause (int ch)

---

Description

Pauses play. Issue to slave assignment data is taken as an error.

Argument

ch	Sequence No (0 to 3)
----	----------------------

Returned value

0	Normal ending
-1	Error

#### 4.11 Phrase\_Restart

int Phrase\_Restart (int ch)

---

Description

Cancels pause of play. Issue to slave assignment data is taken as an error.

Argument

ch	Sequence No (0 to 3)
----	----------------------

Returned value

0	Normal ending
-1	Error

**4.12 Phrase\_Kill**int Phrase\_Kill (void)

Description

Returns to the unused condition of Phrase sequencer.

Argument

None

Returned value

0	Normal ending
-1	Error

**4.13 Phrase\_SetVolume**void Phrase\_SetVolume (int ch,int vol)

Description

Sets volume. It is always effective after initialization.  
The setup is initialized only by Phrase\_Initialize().

Argument

ch	Sequence No (0 to 3)
vol	Value of volume (-1, 0 to 127) Volume [dB] = 40 * Log (vol/127) Default is 100. -1 means that the default value is set.

Returned value

None

**4.14 Phrase\_GetVolume**int Phrase\_GetVolume (int ch)

Description

Acquires volume.

Argument

ch	Sequence No (0 to 3)
----	----------------------

Returned value

Non-negative	Sets volume. (0 to 127)
-1	Error

#### 4.15 Phrase\_SetPanpot

void Phrase\_SetPanpot (int ch,int pan)

Description

Sets panpot. Since this operation is for the same resources as the ones for panpot designation in the SMAF/Phrase sequencer, when both the setting here and setting in the data are present, the one arriving later is given precedence.

The pan pot value in the sequence data head is the set value in this API, and when it returns to the head by repeat, it becomes the set value in this API.

It is always effective after initialization.

The setup is initialized only by Phrase\_Initialize().

Argument

ch	Sequence No (0 to 3)
pan	Panpot (-1, 0 to 127)
	VolL[dB] = 20 * Log (Cos (PAI/2 * pan/127)), MUTE if pan == 127
	VolR[dB] = 20 * Log (Sin (PAI/2 * pan/127)), MUTE if pan == 0
	Default is 64
	-1 means that the default value is set.

Returned value

None

#### 4.16 Phrase\_GetPanpot

int Phrase\_GetPanpot (int ch)

Description

Acquires panpot that is set with Phrase\_SetPanpot ().

Argument

ch	Sequence No (0 to 3)
----	----------------------

Returned value

Non-negative	Panpot setting(0 to 127)
-1	Error

#### 4.17 Phrase\_GetStatus

int Phrase\_GetStatus (int ch)

Description

Acquires status.

Argument

ch	Sequence No (0 to 3)
----	----------------------

Returned value

Lower 4 bits (bit3..0)

1 no data

2 ready

3 play

4 ending

5 pause

5<sup>th</sup> bit from the last one (bit4)

0 Master

1 link-slave

**4.18 Phrase\_RemoveData**int Phrase\_RemoveData (int ch)


---

Description	Registration deletion of SMAF/Phrase data.	
Argument	ch	Sequence No (0 to 3)
Returned value	0	Normal ending
	-1	Error

**4.19 Phrase\_SetEvHandler**int Phrase\_SetEvHandler (void \*func (struct event \*eve))


---

Description	Sets user event and reproduction ending call back. When there is not setting, no call beck is made.	
Argument	func eve	Pointer for call back function Pointer for event structural body struct event{ int ch;//Sequence No (0 to 3) int mode;//mode -0x01 : Ending of reproduction -0x02 : Ending of loop reproduction 0x00 to 0x0F : conforming to user events 0xFF10 to 0xFF1F }
Returned value	0	Normal ending
	-1	Error

#### 4.20 Phrase\_SetLink

int Phrase\_SetLink (int ch,unsigned long slave)

Description

Sets synchronous reproduction sequence.

For the sequence that is in the slave setting, both the master setting of the sequence and slave setting with other master setting cannot be made. the cancellation including all link cancellation by slave=0x00000000 is performed when applicable bit of the same master setting is "0". The cancellation is also performed when the data of master setting is deleted with Phrase\_RemoveData() and slave setting itself is deleted. When all the synchronizing data are not in the ready condition, when the data which become slave are already specified as master or slave, and when all the play time of the synchronizing data is not the same, it becomes an error.

Synchronous operation can be made for five operations, Start, Stop, Pause, Restart, and Seek.

Argument

ch	Sequence No (0 to 3)
slave	Designation of slave sequence bit0 : Sequence No 0 to bit3 : Sequence No 3

Returned value

0	Normal ending
-1	Error

#### 4.21 Phrase\_GetLink

unsigned long Phrase\_GetLink (int ch)

Description

Acquisition of sequence number by which slave setup is carried out to the specified sequence.

Argument

ch	(master)Sequence No (0 to 3)
----	------------------------------

Returned value

Synchronous setup	In the case of 0, no slave setting. bit0 : It is shown that sequence No 0 is slave by "1". bit1 : It is shown that sequence No 1 is slave by "1". bit2 : It is shown that sequence No 2 is slave by "1". bit3 : It is shown that sequence No 3 is slave by "1".
-------------------	---

#### 4.22 Phrase\_GetLength

long Phrase\_GetLength (int ch)

Description

Acquire playback time of the designated sequence. It is valid between Phrase\_SetData and Phrase\_RemoveData.

Argument

ch	Sequence No (0 to 3)
----	----------------------

Returned value

1 or more	Playback time of sequence (msec)
0	Unregistered sequence.