

***Software Synthesizer  
MIDI Player /Driver Library  
Specification***

***Version 3.0***

**CONFIDENTIAL**

***bismark***

**History:**

<b>Date</b>	<b>Version</b>	<b>Description</b>
2013/10/17	2.8	
2014/08/05	3.0	BSMP_CTRL_SET_SAMPLE_RATE および BSMD_CTRL_SET_SAMPLE_RATE の data type を修正

目次

<b>1. 適用 .....</b>	<b>7</b>
<b>2. 概要 .....</b>	<b>7</b>
<b>2.1. 対応 OS .....</b>	<b>9</b>
<b>2.2. 入力 .....</b>	<b>9</b>
<b>2.2.1. MIDI ファイル .....</b>	<b>9</b>
<b>2.2.2. 音色ライブラリファイル .....</b>	<b>9</b>
<b>2.3. 出力 .....</b>	<b>9</b>
<b>2.3.1. Wave 出力デバイス .....</b>	<b>9</b>
<b>2.3.2. Wave ファイル.....</b>	<b>10</b>
<b>2.4. ファイル構成 .....</b>	<b>10</b>
<b>2.5. 関連ライブラリ .....</b>	<b>11</b>
<b>3. MIDI プレーヤライブラリ仕様 .....</b>	<b>12</b>
<b>3.1. 定数 .....</b>	<b>12</b>
<b>3.1.1. BSMP_ERR .....</b>	<b>12</b>
<b>3.1.2. BSMP_CTRL .....</b>	<b>12</b>
<b>3.1.3. BSMP_CALLBACK_TYPE .....</b>	<b>12</b>
<b>3.1.4. BSMP_WAVE_FILE .....</b>	<b>12</b>
<b>3.1.5. BSMP_SOUND_LIBRARY_SEL_MODE .....</b>	<b>13</b>
<b>3.2. 型定義 .....</b>	<b>14</b>
<b>3.2.1. BSMP_HANDLE .....</b>	<b>14</b>
<b>3.2.2. BSMP_CALLBACK .....</b>	<b>14</b>
<b>3.2.3. BSMP_CALLBACK_BOUNCE .....</b>	<b>14</b>
<b>3.2.4. BSMP_LOAD.....</b>	<b>14</b>
<b>3.3. 構造体 .....</b>	<b>14</b>
<b>3.3.1. BSMP_FUNC.....</b>	<b>14</b>
<b>3.3.2. BSMP_SOUND_LIBRARY .....</b>	<b>15</b>
<b>3.3.3. BSMP_SOUND_LIBRARY_MEMORY .....</b>	<b>15</b>
<b>3.3.4. BSMP_SOUND_LIBRARY_SEL.....</b>	<b>15</b>
<b>3.4. API.....</b>	<b>16</b>
<b>3.4.1. initialize.....</b>	<b>16</b>
<b>3.4.2. initializeWithSoundLib .....</b>	<b>17</b>
<b>3.4.3. initializeWithSoundLibMemory .....</b>	<b>17</b>

**CONFIDENTIAL**

3.4.4. <i>exit</i> .....	19
3.4.5. <i>getNumDrivers</i> .....	19
3.4.6. <i>getNumDevices</i> .....	19
3.4.7. <i>getDriverName</i> .....	20
3.4.8. <i>getDeviceName</i> .....	20
3.4.9. <i>showDeviceControlPanel</i> .....	20
3.4.10. <i>open</i> .....	20
3.4.11. <i>close</i> .....	21
3.4.12. <i>setFile</i> .....	22
3.4.13. <i>setFileMemory</i> .....	22
3.4.14. <i>getFileMemory</i> .....	23
3.4.15. <i>getFileInfo</i> .....	23
3.4.16. <i>start</i> .....	24
3.4.17. <i>stop</i> .....	24
3.4.18. <i>seek</i> .....	24
3.4.19. <i>isPlaying</i> .....	25
3.4.20. <i>bounce</i> .....	26
3.4.21. <i>ctrl</i> .....	27
3.4.22. <i>version</i> .....	35
3.5. コールバック ( <i>BSMP_CALLBACK</i> ) .....	36
3.5.1. <i>Open</i> .....	36
3.5.2. <i>Close</i> .....	36
3.5.3. <i>Start</i> .....	36
3.5.4. <i>Stop</i> .....	36
3.5.5. <i>Seek</i> .....	37
3.5.6. <i>MIDI</i> クロック .....	37
3.5.7. <i>Tempo</i> .....	37
3.5.8. <i>Time Signature</i> .....	37
3.5.9. <i>Channel Message</i> .....	37
3.5.10. <i>System Exclusive Message</i> .....	38
3.6. 制御シーケンス .....	39
3.6.1. ライブラリ初期化 .....	39
3.6.2. <i>MIDI</i> ファイル指定 - 演奏開始 - 演奏終了 .....	40
3.6.3. <i>MIDI</i> ファイル指定 - 演奏開始 - 演奏自然終了 .....	41
3.6.4. ライブラリ終了 .....	42

<b>4. MIDI ドライバライブラリ仕様</b>	<b>43</b>
<b>4.1. 定数</b>	<b>43</b>
<b>4.1.1. BSMD_ERR</b>	<b>43</b>
<b>4.1.2. BSMD_CTRL</b>	<b>43</b>
<b>4.1.3. BSMD_CALLBACK_TYPE</b>	<b>43</b>
<b>4.1.4. BSMD_SOUND_LIBRARY_SEL_MODE</b>	<b>43</b>
<b>4.2. 型定義</b>	<b>44</b>
<b>4.2.1. BSMD_HANDLE</b>	<b>44</b>
<b>4.2.2. BSMD_CALLBACK</b>	<b>44</b>
<b>4.2.3. BSMD_LOAD</b>	<b>44</b>
<b>4.3. 構造体</b>	<b>44</b>
<b>4.3.1. BSMD_FUNC</b>	<b>44</b>
<b>4.3.2. BSMD_SOUND_LIBRARY</b>	<b>44</b>
<b>4.3.3. BSMD_SOUND_LIBRARY_MEMORY</b>	<b>44</b>
<b>4.3.4. BSMD_SOUND_LIBRARY_SEL</b>	<b>45</b>
<b>4.3.5. BSMD_FRAME</b>	<b>45</b>
<b>4.4. API</b>	<b>46</b>
<b>4.4.1. initialize</b>	<b>46</b>
<b>4.4.2. initializeWithSoundLib</b>	<b>47</b>
<b>4.4.3. initializeWithSoundLibMemory</b>	<b>47</b>
<b>4.4.4. exit</b>	<b>48</b>
<b>4.4.5. getNumDrivers</b>	<b>48</b>
<b>4.4.6. getNumDevices</b>	<b>48</b>
<b>4.4.7. getDriverName</b>	<b>49</b>
<b>4.4.8. getDeviceName</b>	<b>49</b>
<b>4.4.9. showDeviceControlPanel</b>	<b>49</b>
<b>4.4.10. open</b>	<b>50</b>
<b>4.4.11. close</b>	<b>50</b>
<b>4.4.12. start</b>	<b>50</b>
<b>4.4.13. stop</b>	<b>51</b>
<b>4.4.14. isPlaying</b>	<b>51</b>
<b>4.4.15. setChannelMessage</b>	<b>51</b>
<b>4.4.16. setSystemExclusiveMessage</b>	<b>52</b>
<b>4.4.17. setFile</b>	<b>53</b>
<b>4.4.18. setFileMemory</b>	<b>53</b>

**CONFIDENTIAL**

4.4.19.	getFileMemory .....	54
4.4.20.	getFileInfo .....	54
4.4.21.	startFilePlay.....	55
4.4.22.	stopFilePlay .....	55
4.4.23.	seekFilePlay.....	55
4.4.24.	isFilePlaying.....	56
4.4.25.	ctrl.....	57
4.4.26.	version.....	64
4.5.	コールバック (BSMD_CALLBACK).....	65
4.5.1.	Open.....	65
4.5.2.	Close.....	65
4.5.3.	Start .....	65
4.5.4.	Stop.....	65
4.5.5.	Audio Frame .....	65
4.5.6.	File Start.....	66
4.5.7.	File Stop .....	66
4.5.8.	File Seek .....	66
4.5.9.	MIDI クロック .....	66
4.5.10.	Tempo.....	66
4.5.11.	Time Signature .....	67
4.5.12.	Channel Message .....	67
4.5.13.	System Exclusive Message.....	67
4.6.	制御シーケンス.....	68
4.6.1.	起動および Real-time MIDI 再生の開始 .....	68
4.6.2.	MIDI ファイル指定 - 演奏開始 - 演奏終了 .....	69
4.6.3.	MIDI ファイル指定 - 演奏開始 - 演奏自然終了 .....	70
4.6.4.	ライブラリ終了.....	71
5.	付録.....	72
5.1.	DLS フォーマットの対応について.....	72

## 1. 適用

本仕様書はソフトウェアシンセサイザ MIDI プレーヤ/ドライバライブラリに適用する。

## 2. 概要

本ライブラリは、シンセサイザエンジンライブラリ(bsse: bismark Synthesizer Engine)および音色ライブラリを内蔵し、対アプリケーション用として、MIDI プレーヤ(bsmp: 後述)、ならびに MIDI ドライバ(bsmd: 後述)ライブラリのインターフェースを提供するものである。

bsmp (bismark MIDI Player) ライブラリは、シンセサイザエンジンライブラリ専用の付属ライブラリである。MIDI プレーヤ/KARAOKE、MIDI to Wave コンバータ等のアプリケーション構築を容易とするための機能を提供する。

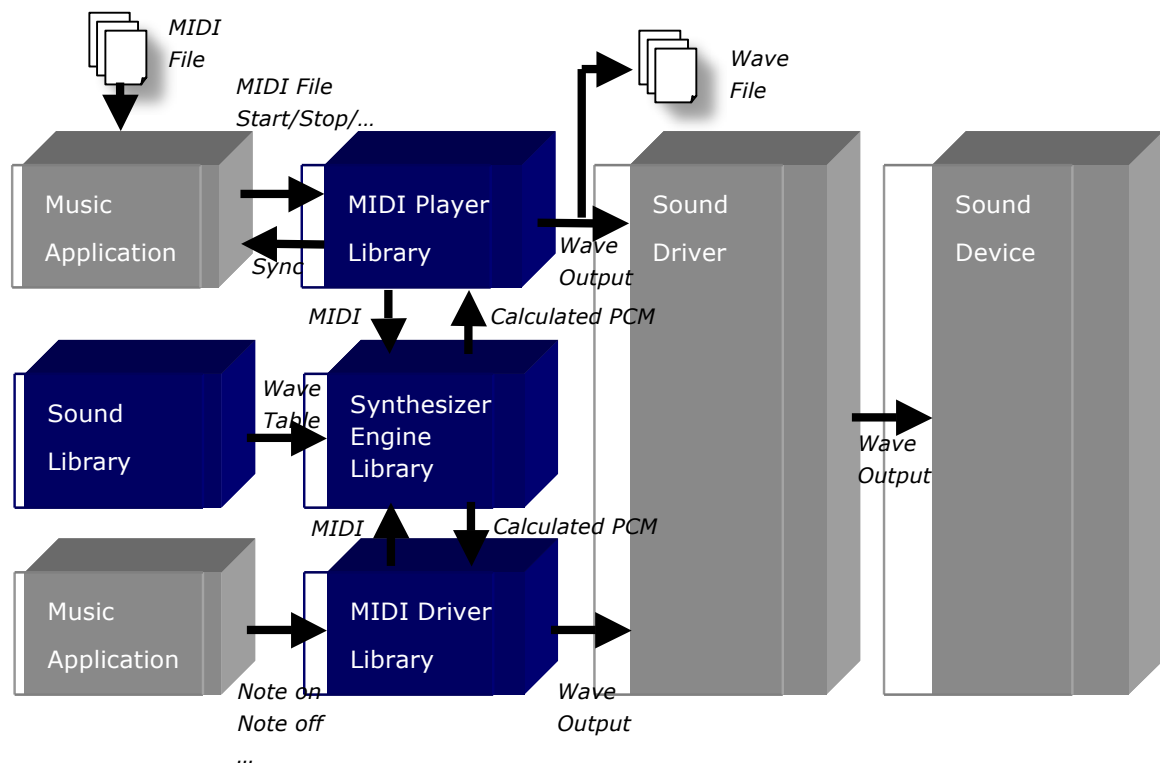
主な基本機能は下記の通りである。

- MIDI ファイル読込
  - SMF(Standard MIDI File)対応
  - ...その他各種形式に別途カスタマイズ対応が可能
- シンセサイザエンジンライブラリを用いた MIDI to Wave 変換
  - 各対応 OS の Wave 出力デバイス管理および Thread Schedule 管理
  - Wave ファイルへの変換出力
- アプリケーション支援
  - 再生、停止等の制御 API
  - コールバック機能によるアプリケーションへの MIDI 演奏同期情報の出力

同じく、bsmd (bismark MIDI Driver) ライブラリは、シンセサイザエンジンライブラリ(bsse: bismark Synthesizer Engine)専用の付属ライブラリである。既存ハードウェア MIDI 音源の代替を可能とし、仮想楽器アプリケーション等に Real-time MIDI 再生および簡易 MIDI ファイルプレーヤ機能を提供する。

主な基本機能は下記の通りである。

- Real-time MIDI 再生機能
  - 各対応 OS の Wave 出力デバイス管理および Thread Schedule 管理を含む
- MIDI ファイル簡易再生機能
  - SMF(Standard MIDI File)対応



MIDI プレーヤライブラリ、および MIDI ドライバライブラリはいずれか一方の排他的呼び出しにのみ対応する。同時使用は保障されない。



## **2.1. 対応 OS**

以下の OS をサポートする。

- Win 版
  - MBCS ビルド
  - UNICODE ビルド
- Linux/BSD 版
- Mac OS X 版
- iOS 版 (iOS 7 SDK, armv7/armv7s)
- Android 版 (NDK r7b)

## **2.2. 入力**

### **2.2.1. MIDI ファイル**

- SMF (Standard MIDI File)
  - Format: 0、および 1
  - Number of track: 最大 64
  - 分解能 / TPQN: 制限無し
  - 拡張子: \*.mid (但し、本ライブラリでは拡張子には関知しない)

### **2.2.2. 音色ライブラリファイル**

- SoundFont
  - Version 2
  - 拡張子: \*.sf2 (但し、本ライブラリでは拡張子には関知しない)
- DLS (Downloadable Sounds)<sup>1</sup>
  - Level1, Level2, Mobile DLS
  - 拡張子: \*.dls (但し、本ライブラリでは拡張子には関知しない)

## **2.3. 出力**

### **2.3.1. Wave 出力デバイス**

- Win 版:

---

<sup>1</sup> 現状ステレオ形式の波形データ等、いくつか DLS 規格において定義される仕様に対して未対応項目があります。詳細は「5.1 DLS フォーマットの対応について」をご参照下さい。

- MME 形式標準ドライバ対応
- Steinberg ASIO 2.1 ドライバ対応 (MIDI ドライバライブラリ、かつ 44100Hz のみ)
- Linux 版:
  - OSS 対応
  - ALSA 対応
- Mac OS X 版、iOS 版:
  - AudioQueue 対応
  - AudioUnit 対応 (MIDI ドライバライブラリのみ)
- Android 版
  - OpenSL ES
- 再生サンプリング周波数: 該当ドライバの対応に依存

### **2.3.2. Wave ファイル**

MIDI プレーヤライブラリのみ下記 Wave ファイルの出力に対応する。

- Microsoft RIFF Wave
- Apple AIFF
  - 再生サンプリング周波数: 任意
  - 出力ビット精度: 16[bit]
  - 出力チャンネル: 2 (Interleaved)

### **2.4. ファイル構成**

- OS 共通
  - bsmd.h : MIDI ドライバライブラリヘッダファイル
  - bsmp.h : MIDI プレーヤライブラリヘッダファイル
- Win 版 (共有ライブラリ)
  - bsmpd\*.dll : 共有ライブラリ
  - bsmpd\*.lib : ライブラリモジュール
- Linux / Mac OS X / iOS / Android 版 (スタティックライブラリ)
  - libbsmpd\*.a (MIDI プレーヤ/MIDI ドライバライブラリ)
  - libbsmp\*.a (MIDI プレーヤライブラリ)
  - libbsmd\*.a (MIDI ドライバライブラリ)

## **2.5. 関連ライブラリ**

- シンセサイザエンジンライブラリ
  - Win 版
    - ◇ 本ライブラリに内蔵
  - Linux / Mac OS X / iOS / Android 版
    - ◇ libbsse\*.a:スタティックライブラリ

**CONFIDENTIAL**

### 3. MIDI プレーヤライブラリ仕様

#### 3.1. 定数

##### 3.1.1. BSMP\_ERR

エラーコードを表す typedef enum。

code	内容	
BSMP_OK	正常終了	
BSMP_ERR_PROTECTION	プロテクションエラー	
BSMP_ERR_INVALID_HANDLE	ハンドル異常	
BSMP_ERR_FILE	ファイル入出力関連エラー	
BSMP_ERR_MEMORY	メモリ取得関連エラー	
BSMP_ERR_RESOURCE	リソース関連エラー	
BSMP_ERR_PARAM	パラメータ関連エラー	
BSMP_ERR_AUDIO_DRIVER	Wave 出カドライバ関連エラー	
BSMP_ERR_DATA	データ関連エラー	
BSMP_ERR_MODULE	外部モジュール関連エラー	
BSMP_ERR_NOT_SUPPORTED	非対応項目等	
BSMP_ERR_UNDEFINED	未定義	

##### 3.1.2. BSMP\_CTRL

制御コードを表す typedef enum。詳細は 3.4.21 ctrl を参照のこと。

##### 3.1.3. BSMP\_CALLBACK\_TYPE

コールバック種別を表す typedef enum。詳細は 3.5 コールバック (BSMP\_CALLBACK) を参照のこと。

##### 3.1.4. BSMP\_WAVE\_FILE

Wave ファイル出力形式をあらわす typedef enum。

code	内容	
BSMP_WAVE_FILE_RIFF	Microsoft RIFF Wave	
BSMP_WAVE_FILE_AIFF	Apple AIFF	

### **3.1.5. BSMP\_SOUND\_LIBRARY\_SEL\_MODE**

音色ライブラリ参照方法を示す typedef enum。

<b>code</b>	<b>内容</b>	
BSMP_SOUND_LIBRARY_SEL_MODE_NORMAL	デフォルト設定	

**CONFIDENTIAL**

## 3.2. 型定義

### 3.2.1. BSMP\_HANDLE

本ライブラリの制御ハンドル。

### 3.2.2. BSMP\_CALLBACK

本ライブラリからユーザーアプリケーションへの各種通知を行なうコールバック関数。以下の形式とする。詳細は 3.5 コールバック (BSMP\_CALLBACK)を参照のこと。

callback ()

Input:	BSMP_HANDLE handle	本ライブラリの有効なハンドル
	BSMP_CALLBACK_TYPE type	コマンド
	void *data	各コマンド用データ
	void *user	ユーザパラメータ
Output:	void	

### 3.2.3. BSMP\_CALLBACK\_BOUNCE

Wave ファイル出力動作時の経過表示等を行なうコールバック関数。以下の形式とする。3.4.20 bounce 実行時に使用する。

BSMP\_CALLBACK\_BOUNCE ()

Input:	int percent	経過 (%)
	void *user	ユーザデータ
Output:	int	0: 通常終了 1: キャンセル

### 3.2.4. BSMP\_LOAD

ライブラリプラグイン API テーブル(BSMP\_FUNC)を取得する外部公開関数型。

## 3.3. 構造体

### 3.3.1. BSMP\_FUNC

ライブラリの API 関数テーブルを表す構造体。メンバ内容については 3.4 API を参照のこと。

**CONFIDENTIAL**

### 3.3.2. BSMP\_SOUND\_LIBRARY

音色ライブラリファイルの指定に使用する。

```
typedef struct {  
    int index; /* 音色ライブラリのインデックス */  
    LPCTSTR path; /* 音色ライブラリファイルのパス */  
} BSMP_SOUND_LIBRARY;
```

### 3.3.3. BSMP\_SOUND\_LIBRARY\_MEMORY

メモリ上に配置された音色ライブラリファイルの指定に使用する。

```
typedef struct {  
    int index; /* 音色ライブラリのインデックス */  
    char *address; /* 音色ライブラリファイルのアドレス */  
    unsigned long *size; /* 音色ライブラリファイルのサイズ */  
} BSMP_SOUND_LIBRARY_MEMORY;
```

### 3.3.4. BSMP\_SOUND\_LIBRARY\_SEL

音色ライブラリの参照先をエンジンのパート毎に指定する

```
typedef struct {  
    int module; /* モジュール番号 (0, 1, ...) */  
    int part; /* パート番号 (0, 1, ..., 15) */  
    int index; /* 音色ライブラリのインデックス */  
    BSMP_SOUND_LIBRARY_SEL_MODE mode; /* 参照方法 (3.1.5 参照) */  
} BSMP_SOUND_LIBRARY_SEL;
```

**CONFIDENTIAL**

### 3.4. API

#### 3.4.1. initialize

BSMP\_ERR initialize ()

Input:

BSMP_HANDLE *handle	本ライブラリのハンドル取得ポインタ (NULL 不可)
BSMP_CALLBACK callback	コールバック関数
void *user	コールバック関数用ユーザパラメータ
void *target	OS 依存パラメータ
const unsigned char *key	起動キー

Output:

エラーコード

本ライブラリ、およびシンセサイザエンジンライブラリを起動する。

既定の音色ライブラリ(インデックス#0)には内蔵、または標準定義される内容が自動的に読み込まれる。

本ライブラリ使用時は必ず最初にこの関数を含む、initialize\* ()関数のいずれかひとつのみを実行するものとし、正常終了するまで他の関数は実行できない。

起動キー(引数 key)には別途提供される 64byte の起動キーの先頭アドレスを指定する。

本関数内で音色ライブラリの読み込み処理を行うため、環境によっては実行時間が必要とされることがある。このため必要な場合にはアプリケーション側で wait cursor 表示等を行うこと。

target に設定するパラメータは下記の通りとする。

- Win 版: 親ウインドウハンドル (HWND)
- Android 版: 本ライブラリは下記構造体のポインタを引数として受け取り、アプリケーションの Activity クラスのメソッドをコールします。

```
typedef struct {  
    JNIEnv *env;  
    jobject thiz;  
}
```

- 上記以外: NULL



**CONFIDENTIAL**

### 3.4.2. initializeWithSoundLib

BSMP\_ERR initializeWithSoundLib ()

Input:

BSMP_HANDLE *handle	本ライブラリのハンドル取得ポインタ (NULL 不可)
BSMP_CALLBACK callback	コールバック関数
void *user	コールバック関数用ユーザパラメータ
LPCTSTR libraryPath	音色ライブラリのパス
void *target	OS 依存パラメータ
const unsigned char *key	起動キー

Output:

エラーコード

本ライブラリ、およびシンセサイザエンジンライブラリを起動する。

既定の音色ライブラリ(インデックス#0)には指定パスの音色ライブラリファイルが適用される。

その他注意事項は initialize の項を参照のこと。

### 3.4.3. initializeWithSoundLibMemory

BSMP\_ERR initializeWithSoundLibMemory ()

Input:

BSMP_HANDLE *handle	本ライブラリのハンドル取得ポインタ (NULL 不可)
BSMP_CALLBACK callback	コールバック関数
void *user	コールバック関数用ユーザパラメータ
char *libraryAddress	音色ライブラリのアドレス
unsigned long librarySize	音色ライブラリのサイズ
void *target	OS 依存パラメータ
const unsigned char *key	起動キー

Output:

エラーコード

本ライブラリ、およびシンセサイザエンジンライブラリを起動する。

既定の音色ライブラリ(インデックス#0)には指定メモリ上の音色ライブラリファイルが適用される。  
その他注意事項は initialize の項を参照のこと。

#### **3.4.4. exit**

BSMP\_ERR exit ()

Input:

BSMP\_HANDLE handle      本ライブラリの有効なハンドル

Output:

エラーコード

本ライブラリを終了する。

本ライブラリ使用時は必ず最後にこの関数を実行する。演奏中の場合には先ず演奏停止処理を行ってから実行すること。

#### **3.4.5. getNumDrivers**

int getNumDrivers ()

Input:

BSMP\_HANDLE handle      本ライブラリの有効なハンドル

Output:

対応する Wave 出力ドライ

バ数

本ライブラリが対応する Wave 出力ドライバ数を取得する。

#### **3.4.6. getNumDevices**

int getNumDevices ()

Input:

BSMP\_HANDLE handle      本ライブラリの有効なハンドル

LPCTSTR driver          Wave 出力ドライバ名

Output:

対応する Wave 出力デバイ

ス数

指定された Wave 出力ドライバ名称において本ライブラリが使用可能なデバイス数を取得する。

**CONFIDENTIAL**

### 3.4.7. getDriverName

LPCTSTR getDriverName ()

Input:

BSMP_HANDLE handle	本ライブラリの有効なハンドル
int index	Wave 出力ドライバのインデックス

Output:

Wave 出力ドライバ名

指定インデックスに相当する Wave 出力ドライバ名を取得する。

### 3.4.8. getDeviceName

LPCTSTR getDeviceName ()

Input:

BSMP_HANDLE handle	本ライブラリの有効なハンドル
LPCTSTR driver	Wave 出力ドライバ名
int index	Wave 出力デバイスのインデックス

Output:

Wave 出力ドライバ名

指定 Wave 出力ドライバにおいて、指定インデックスに相当するデバイス名を取得する。

### 3.4.9. showDeviceControlPanel

void showDeviceControlPanel ()

Input:

BSMP_HANDLE handle	本ライブラリの有効なハンドル
LPCTSTR driver	Wave 出力ドライバ名
LPCTSTR device	Wave 出力デバイス名

指定 Wave 出力ドライバおよびデバイスの制御パネルを表示する。

### 3.4.10. open

BSMP\_ERR open ()

Input:

BSMP_HANDLE handle	本ライブラリの有効なハンドル
LPCTSTR driver	Wave 出力ドライバ名
LPCTSTR device	Wave 出力デバイス名

Output:

エラーコード

指定の Wave 出力ドライバおよびデバイスを open する。

ドライバおよびデバイス名が NULL の場合にはデフォルトドライバならびにデバイスが自動選択される。

#### **3.4.11. close**

BSMP\_ERR close ()

Input:

BSMP_HANDLE handle	本ライブラリの有効なハンドル
--------------------	----------------

Output:

エラーコード

Wave 出力デバイスを close する。

### **3.4.12. setFile**

BSMP\_ERR setFile ()

Input:

BSMP_HANDLE handle	本ライブラリの有効なハンドル
LPCTSTR path	MIDI ファイルのフルパス名

Output:

エラーコード

演奏する MIDI ファイルを指定する。指定可能なファイル format は 2.2 入力に準ずる。

### **3.4.13. setFileMemory**

BSMP\_ERR setFileMemory ()

Input:

BSMP_HANDLE handle	本ライブラリの有効なハンドル
char *address	MIDI ファイルの先頭アドレス
long size	MIDI ファイルのサイズ[byte]

Output:

エラーコード

アプリケーションの管理するメモリ上にロードされている MIDI ファイルを演奏対象として設定する。

#### **3.4.14. getFileMemory**

BSMP\_ERR getFileMemory ()

Input:

BSMP_HANDLE handle	本ライブラリの有効なハンドル
char **address	MIDI ファイルの先頭アドレス
long *size	MIDI ファイルのサイズ[byte]

Output:

エラーコード

本ライブラリの管理するメモリ上にロードされている MIDI ファイルのロードアドレスならびにサイズを取得する。

#### **3.4.15. getFileInfo**

BSMP\_ERR getFileInfo ()

Input:

BSMP_HANDLE handle	本ライブラリの有効なハンドル
int *format	MIDI ファイルの format
unsigned short *division	MIDI ファイルの分解能[TPQN]
unsigned long *totaltick	MIDI ファイルの総 tick 数
unsigned long *totaltime	MIDI ファイルの総演奏時間[s]

Output:

エラーコード

設定された MIDI ファイルの情報を取得する。

### **3.4.16. start**

BSMP\_ERR start ()

Input:

BSMP\_HANDLE handle      本ライブラリの有効なハンドル

Output:

エラーコード

MIDI ファイルの演奏を現在設定されている演奏位置から開始する。

### **3.4.17. stop**

BSMP\_ERR stop ()

Input:

BSMP\_HANDLE handle      本ライブラリの有効なハンドル

Output:

エラーコード

現在行なわれている MIDI ファイルの演奏を停止(途中終了)する。

本関数は終了処理の開始(フェードアウト処理の開始)を指示するものであり、本関数終了時点での終了処理の完了は保証されない。実際の演奏終了処理の完了は、後述するコールバックによる通知が行なわれた時点となる。

本関数による演奏停止後も演奏位置は保持される。

### **3.4.18. seek**

BSMP\_ERR seek ()

Input:

BSMP\_HANDLE handle      本ライブラリの有効なハンドル

unsigned long tick      演奏位置 [MIDI tick 単位]

Output:

エラーコード

MIDI ファイルの演奏位置を設定する。



### **3.4.19. isPlaying**

```
int isPlaying ()
```

Input:

BSMP\_HANDLE handle      本ライブラリの有効なハンドル

Output:

1: 演奏中

0: 非演奏中

MIDI ファイルの演奏中であるかを取得する。

### **3.4.20. bounce**

BSMP\_ERR bounce ()

Input:

BSMP_HANDLE handle	本ライブラリの有効なハンドル
LPCTSTR path	出力ファイルのフルパス名
BSMP_WAVE_FILE type	出力ファイル形式
BSMP_CALLBACK_EXPORT callback	経過処理用コールバック関数
void *user	同コールバック関数のユーザーパラメータ

Output:

エラーコード

MIDI ファイルの演奏結果を Wave ファイルに変換出力する。出力完了まで本 API は終了しない。

**3.4.16 start** による演奏が行なわれている状態で本 API を実行することは禁止とする。出力されるファイルのサンプリング周波数は本ライブラリにおける設定値に従うが、出力チャンネル数はライブラリに依存し、ビット精度は 16[bit]固定とする。

### **3.4.21. ctrl**

BSMP\_ERR ctrl ()

Input:

BSMP_HANDLE handle	本ライブラリの有効なハンドル
BSMP_CTRL ctrl	制御対象
void *data	制御データのアドレス
int size	制御データのサイズ[byte]

Output:

エラーコード

各種制御値の設定、取得を行う。制御対象（ctrl）毎に下記に詳細を述べる。

ctrl	data		内容
	type	I/O	
BSMP_CTRL_SET_MASTER_VOLUME	int	I	再生音量(BSMP_VOLUME_MIN～BSMP_VOLUME_MAX)を設定する。初期設定値は BSMP_VOLUME_DEF とする。
BSMP_CTRL_GET_MASTER_VOLUME	int	O	再生音量を取得する。
BSMP_CTRL_SET_MASTER_KEY	int	I	再生キー(BSMP_KEY_MIN～BSMP_KEY_MAX)を設定する。動作は 100[cent]単位とし、初期設定値は BSMP_KEY_DEF とする。演奏開始および終了前後においても設定値は保持されるため、リセット処理はアプリケーション側で実装すること。
BSMP_CTRL_GET_MASTER_KEY	int	O	再生キーを取得する。
BSMP_CTRL_SET_MASTER_TUNE	int	I	再生時のファインチューニング(BSMP_TUNE_MIN～BSMP_TUNE_MAX)を設定する。動作は 1[cent]単位とし初期設定値は BSMP_TUNE_DEF とする。演奏開始および終了前後においても設定値は保持されるため、リセット処理はアプリケーション側で実装すること。
BSMP_CTRL_GET_MASTER_TUNE	int	O	再生時のファインチューニング値を取得する。
BSMP_CTRL_SET_SPEED	int	I	再生スピード(BSMP_SPEED_MIN～BSMP_SPEED_MAX)を設定する。動作は MIDI ファイルに設定されている演奏スピードの 1 [%]単位とし、初期設定値は BSMP_SPEED_DEF とする。演奏開始および終了前後においても設定値は保持されるため、リセット処理はアプリケーション側で実装すること。
BSMP_CTRL_GET_SPEED	int	O	再生スピードを取得する

ctrl	data		内容
	type	I/O	
BSMP_CTRL_SET_GUIDE	int	I	MIDI ファイルのガイドメロディ再生音量 (BSMP_GUIDE_MIN～BSMP_GUIDE_MAX)を設定する。初期設定値は BSMP_GUIDE_DEF とする。演奏開始および終了前後においても設定値は保持されるため、リセット処理はアプリケーション側で実装すること。
BSMP_CTRL_GET_GUIDE	int	O	ガイドメロディ再生音量を取得する。
BSMP_CTRL_SET_GUIDE_MAIN_CH	int	I	ガイドメロディ再生音量制御の対象とする MIDI ポート/チャンネルを設定する。設定値には ポート(0, 1, ...) × 16 + チャンネル (0, 1, ...15) を設定する。設定対象を設けない場合には-1を設定すること。
BSMP_CTRL_GET_GUIDE_MAIN_CH	int	O	ガイドメロディ再生音量制御対象を取得する。
BSMP_CTRL_SET_GUIDE_SUB_CH	int	I	BSMP_CTRL_SET_GUIDE_MAIN_CH と同様。(デュエット等に使用する)
BSMP_CTRL_GET_GUIDE_SUB_CH	int	O	BSMP_CTRL_GET_GUIDE_MAIN_CH と同様。(デュエット等に使用する)

ctrl	data		内容
	type	I/O	
BSMP_CTRL_SET_REVERB	int	I	シンセサイザエンジンライブラリのリバーブ効果の有無を設定する。演奏開始および終了前後においても設定値は保持される。
BSMP_CTRL_GET_REVERB	int	O	シンセサイザエンジンライブラリのリバーブ効果の有無を取得する。
BSMP_CTRL_GET_REVERB_AVAILABLE	int	O	シンセサイザエンジンライブラリがリバーブ効果に対応しているかを取得する。
BSMP_CTRL_SET_CHORUS	int	I	シンセサイザエンジンライブラリのコーラス効果の有無を設定する。演奏開始および終了前後においても設定値は保持される。
BSMP_CTRL_GET_CHORUS	int	O	シンセサイザエンジンライブラリのコーラス効果の有無を取得する。
BSMP_CTRL_GET_CHORUS_AVAILABLE	int	O	シンセサイザエンジンライブラリがコーラス効果に対応しているかを取得する。
BSMP_CTRL_SET_DELAY	int	I	シンセサイザエンジンライブラリのディレイ効果の有無を設定する。演奏開始および終了前後においても設定値は保持される。
BSMP_CTRL_GET_DELAY	int	O	シンセサイザエンジンライブラリのディレイ効果の有無を取得する。
BSMP_CTRL_GET_DELAY_AVAILABLE	int	O	シンセサイザエンジンライブラリがディレイ効果に対応しているかを得る。
BSMP_CTRL_SET_REVERB_HQ	int	I	HQ リバーブを有効にする(1: On, 0: Off、カスタマイズバージョンのみ)

ctrl	data		内容
	type	I/O	
BSMP_CTRL_SET_SAMPLE_RATE	int	I	再生サンプリング周波数を設定する。3.4.10 open により wave 出力デバイスが open されていない状態でのみ設定可能である。
BSMP_CTRL_GET_SAMPLE_RATE	int	O	再生サンプリング周波数[Hz]を取得する。
BSMP_CTRL_SET_BLOCK_SIZE	long	I	再生処理フレームサイズ[sample]を設定する。 一般には外部操作しないこと。
BSMP_CTRL_GET_BLOCK_SIZE	long	O	再生処理フレームサイズ[sample]を取得する。
BSMP_CTRL_SET_CHANNELS	int	I	未サポート
BSMP_CTRL_GET_CHANNELS	int	O	出力チャンネル数を取得する
BSMP_CTRL_SET_POLY	int	I	シンセサイザの最大同時発音数を設定する。演奏中の変更は非対応とする。
BSMP_CTRL_GET_POLY	int	O	シンセサイザの最大同時発音数を取得する。

ctrl	data		内容
	type	I/O	
BSMP_CTRL_GET_SOUND_LIBRARY_NUM	int	O	音色ライブラリの最大格納数を取得する
BSMP_CTRL_SET_SOUND_LIBRARY	BSMP_SOUND_LIBRARY	I	ファイルより音色ライブラリを設定する
BSMP_CTRL_SET_SOUND_LIBRARY_MEMORY	BSMP_SOUND_LIBRARY_MEMORY	I	メモリより音色ライブラリを設定する
BSMP_CTRL_SET_SOUND_LIBRARY_SEL	BSMP_SOUND_LIBRARY_SEL	I	音色ライブラリの参照先を設定する
BSMP_CTRL_GET_SOUND_LIBRARY_SEL	BSMP_SOUND_LIBRARY_SEL	I/O	音色ライブラリの参照内容を取得する
BSMP_CTRL_SET_NO_INSTRUMENT_FIX	int	I	該当音色が無い場合の代替機能を有効にする (1:On, 0: Off)
BSMP_CTRL_GET_NO_INSTRUMENT_FIX	int	O	上記機能の設定値を取得する
BSMP_CTRL_SET_NUMBER_OF_REGIONS	int	I	音色あたりの最大 Region 数を設定する。



ctrl	data		内容
	type	I/O	
BSMP_CTRL_GET_INSTRUMENT_NAME ~ BSMP_CTRL_GET_INSTRUMENT_NAME + 15	char (TCHAR)	O	指定パート(Ch1~16)の現在の音色名を取得する
BSMP_CTRL_SET_MUTE ~ BSMP_CTRL_SET_MUTE + 15	int	I	指定パート(Ch1~16)のミュート設定を行う (0: Off, 1: On)
BSMP_CTRL_GET_MUTE ~ BSMP_CTRL_GET_MUTE + 15	int	O	指定パート(Ch1~16)のミュート設定を取得する (0: Off, 1: On)
BSMP_CTRL_SET_SOLO ~ BSMP_CTRL_SET_SOLO + 15	int	I	指定パート(Ch1~16)のソロ設定を行う (0: Off, 1: On)
BSMP_CTRL_GET_SOLO ~ BSMP_CTRL_GET_SOLO + 15	int	O	指定パート(Ch1~16)のソロ設定を取得する (0: Off, 1: On)

ctrl	data		内容
	type	I/O	
BSMP_CTRL_SET_CAL LBACK_DELAY	int	I	コールバック同期時間オフセットを設定する
BSMP_CTRL_GET_CAL LBACK_DELAY	int	O	コールバック同期時間オフセットを取得する
BSMP_CTRL_SET_POR T_SELECTION_METHO D	int	I	ポート判別方法を設定する(カスタマイズバージョンのみ)
BSMP_CTRL_GET_POR T_SELECTION_METHO D	int	O	ポート判別方法を取得する(カスタマイズバージョンのみ)
BSMP_CTRL_SET_WA VE	BSMP_WAVE	I	Wave ファイルを追加する(カスタマイズバージョンのみ)
BSMP_CTRL_GET_OPE N_SL_ENGINE		O	OpenSL エンジンを取得する(Android のみ)
CTMP_CTRL_GET_OPE N_SL_ENGINE_INTER FACE		O	OpenSL エンジンインターフェースを取得する(Android のみ)

### **3.4.22. version**

```
void version ()
```

Input:

BSMP_HANDLE handle	本ライブラリの有効なハンドル
LPTSTR engine	シンセサイザエンジンライブラリのバージョン文字列
int engineSize	シンセサイザエンジンライブラリのバージョン文字数
LPTSTR player	本ライブラリバージョン文字列
int playerSize	本ライブラリバージョン文字数

Output:

```
void
```

本ライブラリならびにシンセサイザエンジンライブラリのバージョン情報を得る。

### **3.5. コールバック (BSMP\_CALLBACK)**

アプリケーションへ演奏中の各種情報を提供する。コールバック関数は 3.2.2 BSMP\_CALLBACK に示す形式とし、コールバック用のユーザーパラメータと共に、3.4.1 initialize によって設定する。

本コールバックは Wave 変換出力 (3.4.20 bounce) 時はコールされない。

各コールバックはシンセサイザの演算スレッドから呼び出される。従って、シンセサイザの動作へ影響を及ぼすことがないよう、各コールバック時のアプリケーション側の処理は必要最小限とし、処理時間を要するものに関してはコールバックはトリガとしての使用に留め、実処理自体は(カラオケ歌詞文字演算/表示処理等)は別スレッドにて行うよう配慮すること。

#### **3.5.1. Open**

type = BSMP\_CALLBACK\_TYPE\_OPEN, data = 未使用

Wave 出力ドライバがオープンされたことを示す。

#### **3.5.2. Close**

type = BSMP\_CALLBACK\_TYPE\_CLOSE, data = 未使用

Wave 出力ドライバがクローズされたことを示す。

#### **3.5.3. Start**

type = BSMP\_CALLBACK\_TYPE\_START, data = 未使用

演奏開始を示す。楽曲冒頭からの演奏開始を行う場合にはレイテンシを考慮し、楽曲演奏時間先頭に該当する処理(MIDI ファイル上の 0[tick]に相当する処理、カラオケ歌詞文字のタイトル画面の表示等)は本コールバック以降に呼び出される最初のコールバック(MIDI クロック)にて相当する処理が動作するよう配慮すること。

#### **3.5.4. Stop**

type = BSMP\_CALLBACK\_TYPE\_STOP, data = (unsigned long \*) errorcode

演奏終了処理の完了(フェードアウト処理の完了)を示す。本コールバックが呼ばれたスレッドにて次曲の演奏開始処理を行ってはならない。

errorcode は以下の演奏終了状態を示す。

0 : 演奏正常終了

BSMP\_ERR\_AUDIO\_DRIVER : オーディオ Output: ドライバによる演奏異常終了

BSMP\_ERR\_DATA : データ異常による演奏異常終了

### **3.5.5. Seek**

type = BSMP\_CALLBACK\_TYPE\_SEEK, data = 未使用

演奏位置が変更されたことを示す。

アプリケーションにおいて 3.5.6 MIDI クロック を使用して演奏位置算出を行っている場合には、本コールバック受信時に演奏位置を先頭に、テンポ情報を 120[BPM]にリセットすること。

### **3.5.6. MIDI クロック**

type = BSMP\_CALLBACK\_TYPE\_CLOCK, data = 未使用

MIDI ファイルの演奏における MIDI クロックに相当するタイミング更新を示す。

### **3.5.7. Tempo**

type = BSMP\_CALLBACK\_TYPE\_TEMPO, data = (unsigned long \*) tempo

MIDI ファイルの演奏におけるテンポ設定[usec/beat]の変更を示す。

### **3.5.8. Time Signature**

type = BSMP\_CALLBACK\_TYPE\_TIME\_SIGNATURE, data = (unsigned long \*) timeSignature

MIDI ファイルの演奏における拍子設定(nn/dd/cc/bb)の変更を示す。

### **3.5.9. Channel Message**

type = BSMP\_CALLBACK\_TYPE\_CHANNEL\_MESSAGE, data = (unsigned long \*) data

MIDI ファイルの演奏における Channel Message 送信を示す。

bit 31-24: MIDI Port (0x00 ~ )

bit 23 - 16: Status Byte (0x90 ~ 0xEF)

bit 15 - 8 : First Data (0x00 ~ 0x7F)

bit 7 - 0 : Second Data (0x00 ~ 0x7F)

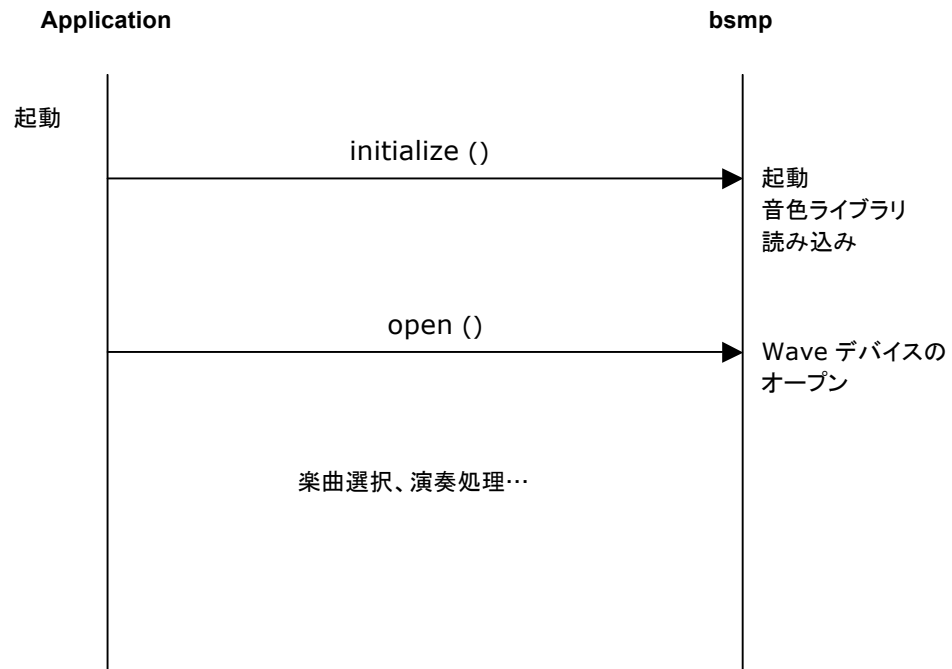
### **3.5.10. System Exclusive Message**

type = BSMP\_CALLBACK\_TYPE\_SYSTEM\_EXCLUSIVE\_MESSAGE, data = 未使用

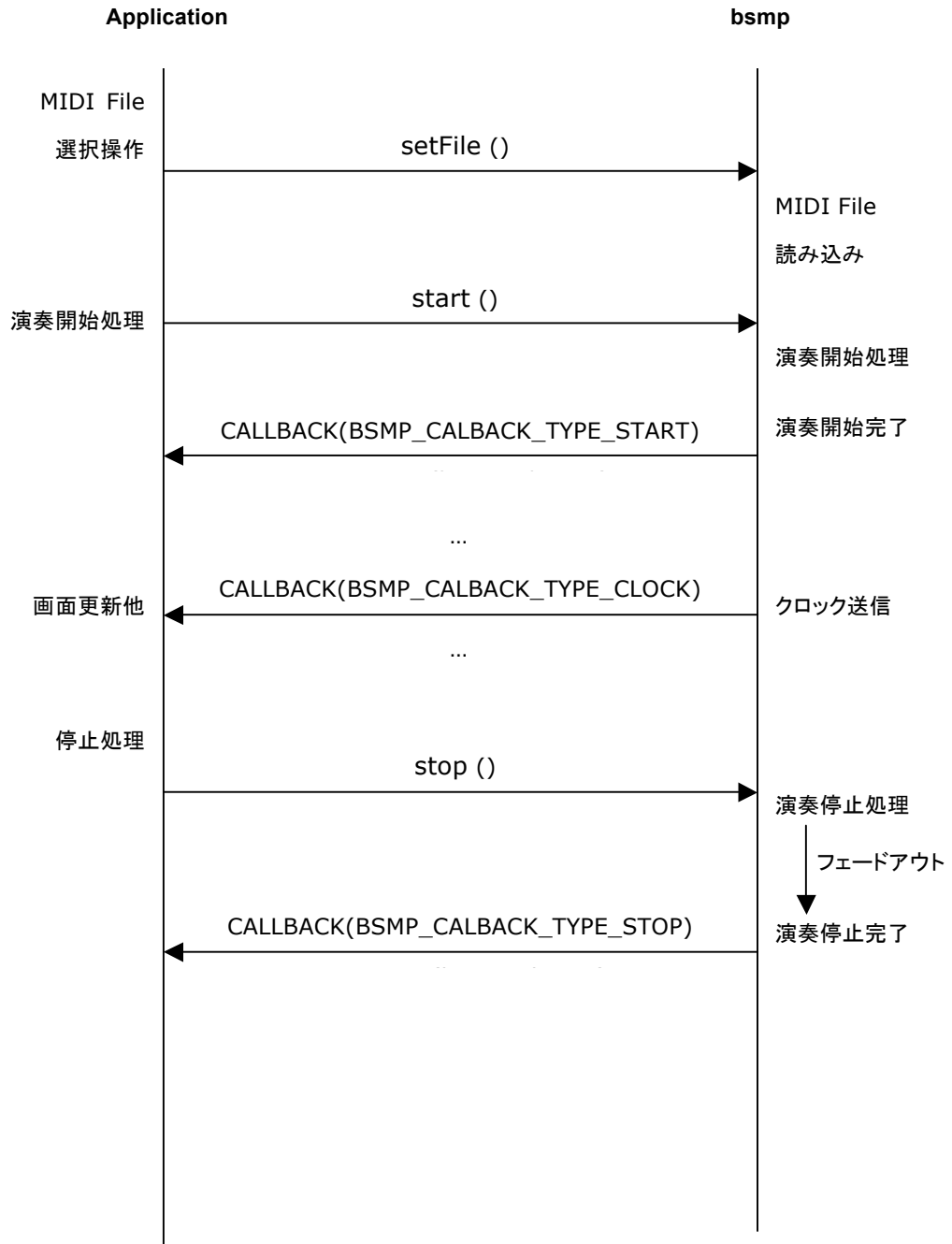
MIDI ファイルの演奏における System Exclusive Message 送信を示す。

### 3.6. 制御シーケンス

#### 3.6.1. ライブラリ初期化

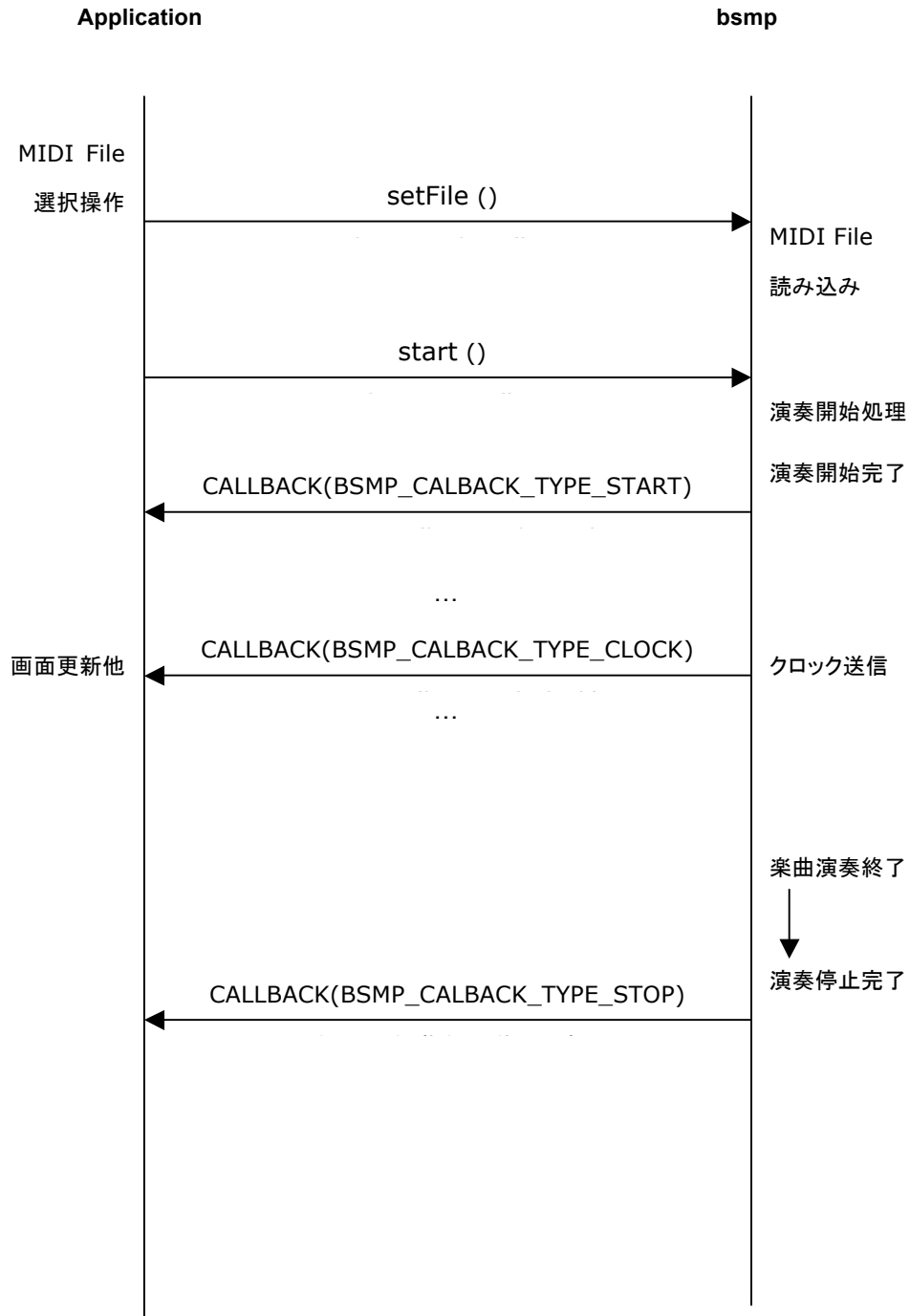


### 3.6.2. MIDI ファイル指定 - 演奏開始 - 演奏終了



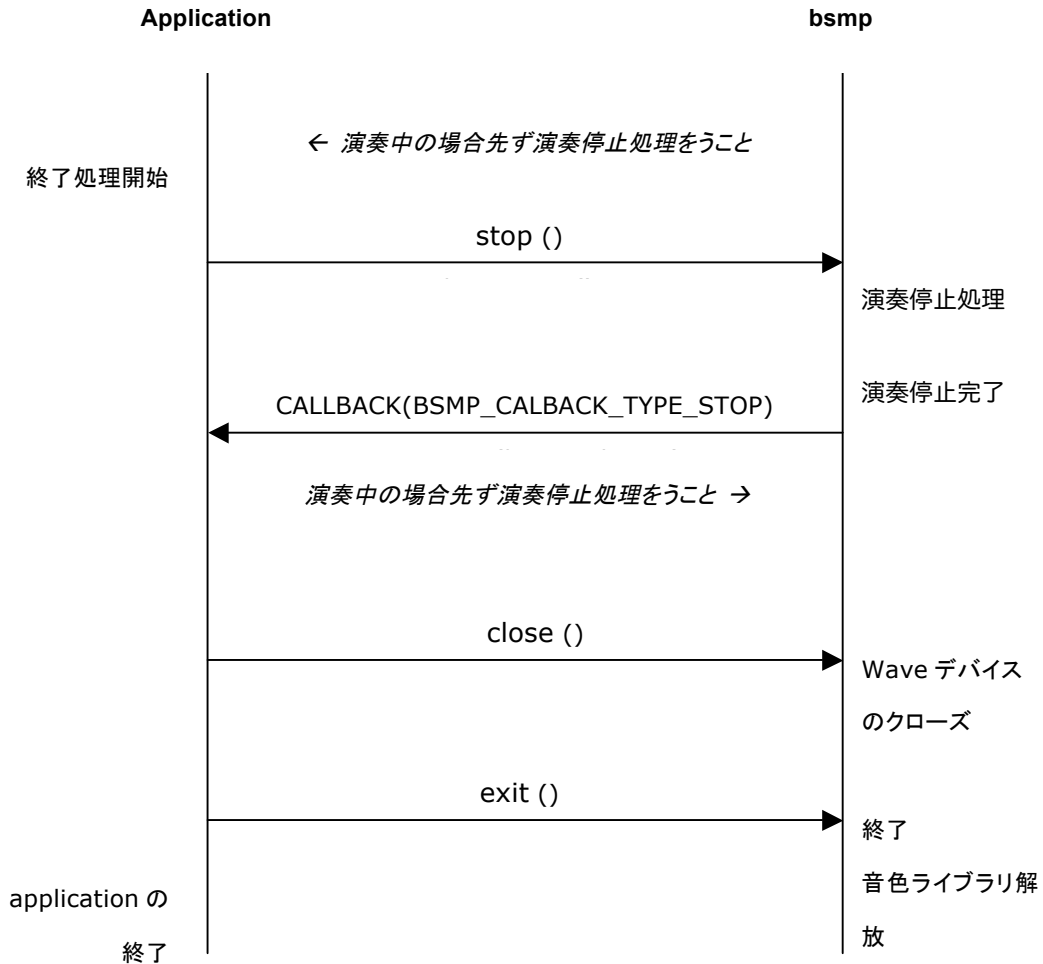


### 3.6.3. MIDI ファイル指定 - 演奏開始 - 演奏自然終了



## 3.6.4. ライブラリ終了

楽曲選択、演奏処理…



演奏中に終了処理を行う場合には先ず演奏停止処理を行ってから Wave 出力デバイスのクローズ、およびライブラリの終了を行う。exit () を実行以後は登録されているコールバック関数はすべて呼び出されない。

**CONFIDENTIAL**

## 4. MIDIドライバライブラリ仕様

### 4.1. 定数

#### 4.1.1. BSMD\_ERR

エラーコードを表す typedef enum。

code	内容	
BSMD_OK	正常終了	
BSMD_ERR_PROTECTION	プロテクションエラー	
BSMD_ERR_INVALID_HANDLE	ハンドル異常	
BSMD_ERR_FILE	ファイル入出力関連エラー	
BSMD_ERR_MEMORY	メモリ取得関連エラー	
BSMD_ERR_RESOURCE	リソース関連エラー	
BSMD_ERR_PARAM	パラメータ関連エラー	
BSMD_ERR_AUDIO_DRIVER	Wave 出力ドライバ関連エラー	
BSMD_ERR_DATA	データ関連エラー	
BSMD_ERR_MODULE	外部モジュール関連エラー	
BSMD_ERR_NOT_SUPPORTED	非対応項目等	
BSMD_ERR_UNDEFINED	未定義	

#### 4.1.2. BSMD\_CTRL

制御コードを表す typedef enum。詳細は 4.4.25 ctrl を参照のこと。

#### 4.1.3. BSMD\_CALLBACK\_TYPE

コールバック種別を表す typedef enum。詳細は 4.5 コールバック (BSMD\_CALLBACK)を参照のこと。

#### 4.1.4. BSMD\_SOUND\_LIBRARY\_SEL\_MODE

音色ライブラリ参照方法を示す typedef enum。

code	内容	
BSMD_SOUND_LIBRARY_SEL_MODE_NORMAL	デフォルト設定	

**CONFIDENTIAL**

## 4.2. 型定義

### 4.2.1. BSMD\_HANDLE

本ライブラリの制御ハンドル。

### 4.2.2. BSMD\_CALLBACK

本ライブラリからユーザーアプリケーションへの各種通知を行なうコールバック関数。以下の形式とする。詳細は 4.5 コールバック (BSMD\_CALLBACK)を参照のこと。

```
BSMD_CALLBACK ()
    Input:  BSMD_HANDLE handle      本ライブラリの有効なハンドル
           BSMD_CALLBACK_TYPE type  コマンド
           void *data               各コマンド用データ
           void *user               ユーザパラメータ
    Output: void
```

### 4.2.3. BSMD\_LOAD

ライブラリプラグイン API テーブル(BSMP\_FUNC)を取得する外部公開関数型。

## 4.3. 構造体

### 4.3.1. BSMD\_FUNC

ライブラリの API 関数テーブルを表す構造体。メンバ内容については 4.4 API を参照のこと。

### 4.3.2. BSMD\_SOUND\_LIBRARY

音色ライブラリファイルの指定に使用する。

```
typedef struct {
    int index; /* 音色ライブラリのインデックス */
    LPCTSTR path; /* 音色ライブラリファイルのパス */
} BSMD_SOUND_LIBRARY;
```

### 4.3.3. BSMD\_SOUND\_LIBRARY\_MEMORY

メモリ上に配置された音色ライブラリファイルの指定に使用する。

**CONFIDENTIAL**

```
typedef struct {  
    int index; /* 音色ライブラリのインデックス */  
    char *address; /* 音色ライブラリファイルのアドレス */  
    unsigned long *size; /* 音色ライブラリファイルのサイズ */  
} BSMD_SOUND_LIBRARY_MEMORY;
```

#### 4.3.4. BSMD\_SOUND\_LIBRARY\_SEL

音色ライブラリの参照先をエンジンのパート毎に指定する

```
typedef struct {  
    int module; /* モジュール番号(0, 1, ...) */  
    int part; /* パート番号 (0, 1, ..., 15) */  
    int index; /* 音色ライブラリのインデックス */  
    BSMD_SOUND_LIBRARY_SEL_MODE mode; /* 参照方法 (4.1.4 参照) */  
} BSMD_SOUND_LIBRARY_SEL;
```

#### 4.3.5. BSMD\_FRAME

コールバック(BSMD\_CALLBACK\_TYPE\_FRAME)時のパラメータに使用する

```
typedef struct {  
    long sampleFrames; /* フレームのサンプル長 */  
    void *data; /* オーディオ出力バッファ (Signed 16bit, 2ch interleaved) */  
} BSMD_FRAME;
```

**CONFIDENTIAL**

## 4.4. API

### 4.4.1. initialize

BSMD\_ERR initialize ()

Input:

BSMD_HANDLE *handle	本ライブラリのハンドル取得ポインタ (NULL 不可)
BSMD_CALLBACK callback	コールバック関数
void *user	コールバック関数用ユーザパラメータ
void *target	OS 依存パラメータ
const unsigned char *key	起動キー

Output:

エラーコード

本ライブラリ、およびシンセサイザエンジンライブラリを起動する。

既定の音色ライブラリ(インデックス #0)には内蔵、または標準定義される内容が自動的に読み込まれる。

本ライブラリ使用時は必ず最初にこの関数を含む、initialize\* ()関数のいずれかひとつのみを実行するものとし、正常終了するまで他の関数は実行できない。

起動キー(引数 key)には別途提供される 64byte の起動キーの先頭アドレスを指定する。

本関数内で音色ライブラリの読み込み処理を行うため、環境によっては実行時間が必要とされることがある。このため必要な場合にはアプリケーション側で wait cursor 表示等を行うこと。

target に設定するパラメータは下記の通りとする。

- Win/WinCE 版: 親ウインドウハンドル (HWND)
- Android 版: 本ライブラリは下記構造体のポインタを引数として受け取り、アプリケーションの Activity クラスのメソッドをコールします。

```
typedef struct {
    JNIEnv *env;
    jobject thiz;
}
```

- 上記以外: NULL

**CONFIDENTIAL**

#### 4.4.2. initializeWithSoundLib

BSMD\_ERR initializeWithSoundLib ()

Input:

BSMD_HANDLE *handle	本ライブラリのハンドル取得ポインタ (NULL 不可)
BSMD_CALLBACK callback	コールバック関数
void *user	コールバック関数用ユーザパラメータ
LPCTSTR libraryPath	音色ライブラリのパス
void *target	OS 依存パラメータ
const unsigned char *key	起動キー

Output:

エラーコード

本ライブラリ、およびシンセサイザエンジンライブラリを起動する。

既定の音色ライブラリ(インデックス#0)には指定パスの音色ライブラリファイルが適用される。

その他注意事項は initialize の項を参照のこと。

#### 4.4.3. initializeWithSoundLibMemory

BSMD\_ERR initializeWithSoundLibMemory ()

Input:

BSMD_HANDLE *handle	本ライブラリのハンドル取得ポインタ (NULL 不可)
BSMD_CALLBACK callback	コールバック関数
void *user	コールバック関数用ユーザパラメータ
char *libraryAddress	音色ライブラリのアドレス
unsigned long librarySize	音色ライブラリのサイズ
void *target	OS 依存パラメータ
const unsigned char *key	起動キー

Output:

エラーコード

本ライブラリ、およびシンセサイザエンジンライブラリを起動する。

既定の音色ライブラリ(インデックス#0)には指定メモリ上の音色ライブラリファイルが適用される。

その他注意事項は initialize の項を参照のこと。

#### **4.4.4. exit**

BSMD\_ERR exit ()

Input:

BSMD\_HANDLE handle      本ライブラリの有効なハンドル

Output:

エラーコード

本ライブラリを終了する。

本ライブラリ使用時は必ず最後にこの関数を実行する。Real-time MIDI 機能再生中の場合には先ず停止処理を行ってから実行すること。

#### **4.4.5. getNumDrivers**

int getNumDrivers ()

Input:

BSMD\_HANDLE handle      本ライブラリの有効なハンドル

Output:

対応する Wave 出力ドライ  
バ数

本ライブラリが対応する Wave 出力ドライバ数を取得する。

#### **4.4.6. getNumDevices**

int getNumDevices ()

Input:

BSMD\_HANDLE handle      本ライブラリの有効なハンドル

LPCTSTR driver          Wave 出力ドライバ名

Output:

対応する Wave 出力デバイ  
ス数



指定された Wave 出力ドライバ名称において本ライブラリが使用可能なデバイス数を取得する。

#### **4.4.7. getDriverName**

LPCTSTR getDriverName ()

Input:

BSMD_HANDLE handle	本ライブラリの有効なハンドル
int index	Wave 出力ドライバのインデックス

Output:

Wave 出力ドライバ名

指定インデックスに相当する Wave 出力ドライバ名を取得する。

#### **4.4.8. getDeviceName**

LPCTSTR getDeviceName ()

Input:

BSMD_HANDLE handle	本ライブラリの有効なハンドル
LPCTSTR driver	Wave 出力ドライバ名
int index	Wave 出力デバイスのインデックス

Output:

Wave 出力ドライバ名

指定 Wave 出力ドライバにおいて、指定インデックスに相当するデバイス名を取得する。

#### **4.4.9. showDeviceControlPanel**

void showDeviceControlPanel ()

Input:

BSMD_HANDLE handle	本ライブラリの有効なハンドル
LPCTSTR driver	Wave 出力ドライバ名
LPCTSTR device	Wave 出力デバイス名

Output:

Wave 出力ドライバ名

指定 Wave 出力ドライバおよびデバイスの制御パネルを表示する。

#### **4.4.10. open**

BSMD\_ERR open ()

Input:

BSMD_HANDLE handle	本ライブラリの有効なハンドル
LPCTSTR driver	Wave 出力ドライバ名
LPCTSTR device	Wave 出力デバイス名

Output:

エラーコード

指定の Wave 出力ドライバおよびデバイスを open する。

ドライバおよびデバイス名が NULL の場合にはデフォルトドライバならびにデバイスが自動選択される。

#### **4.4.11. close**

BSMD\_ERR close ()

Input:

BSMD_HANDLE handle	本ライブラリの有効なハンドル
--------------------	----------------

Output:

エラーコード

Wave 出力デバイスを close する。

#### **4.4.12. start**

BSMD\_ERR start ()

Input:

BSMD_HANDLE handle	本ライブラリの有効なハンドル
--------------------	----------------

Output:

エラーコード

Real-time MIDI 再生機能を開始する。

#### **4.4.13. stop**

BSMD\_ERR stop ()

Input:

BSMD\_HANDLE handle      本ライブラリの有効なハンドル

Output:

エラーコード

Real-time MIDI 再生機能を停止する。

#### **4.4.14. isPlaying**

int isPlaying ()

Input:

BSMD\_HANDLE handle      本ライブラリの有効なハンドル

Output:

0: 停止中

1: 再生中

Real-time MIDI 再生機能の使用状態を取得する。

#### **4.4.15. setChannelMessage**

void setChannelMessage ()

Input:

BSMD\_HANDLE handle      本ライブラリの有効なハンドル

unsigned char port      MIDI ポート (0 = A, 1 = B, ...)

unsigned char status      MIDI ステータス (0x80~0xEF)

unsigned char data1      第 1 データ (0x00~0x7F)

unsigned char data2      第 2 データ (0x00~0x7F)

MIDI チャンネルメッセージを設定する。Real-time MIDI 再生中にのみ使用すること。

#### **4.4.16. setSystemExclusiveMessage**

```
void setSystemExclusiveMessage ()
```

Input:

BSMD_HANDLE handle	本ライブラリの有効なハンドル
unsigned char port	MIDI ポート (0 = A, 1 = B, ...)
unsigned char status	MIDI ステータス (0xF0)
unsigned char *data	データ列の先頭アドレス
int size	データ列長 (Byte)

MIDI システムエクスクルーシブメッセージを設定する。Real-time MIDI 再生中にのみ使用すること。

#### **4.4.17. setFile**

BSMD\_ERR setFile ()

Input:

BSMD_HANDLE handle	本ライブラリの有効なハンドル
LPCTSTR path	MIDI ファイルのフルパス名

Output:

エラーコード

演奏する MIDI ファイルを指定する。指定可能なファイル format は 2.2 入力に準ずる。

#### **4.4.18. setFileMemory**

BSMD\_ERR setFileMemory ()

Input:

BSMD_HANDLE handle	本ライブラリの有効なハンドル
char *address	MIDI ファイルの先頭アドレス
long size	MIDI ファイルのサイズ[byte]

Output:

エラーコード

アプリケーションの管理するメモリ上にロードされている MIDI ファイルを演奏対象として設定する。

#### **4.4.19. getFileMemory**

BSMD\_ERR getFileMemory ()

Input:

BSMD_HANDLE handle	本ライブラリの有効なハンドル
char **address	MIDI ファイルが先頭アドレス
long *size	MIDI ファイルのサイズ[byte]

Output:

エラーコード

本ライブラリの管理するメモリ上にロードされている MIDI ファイルのロードアドレスならびにサイズを取得する。

#### **4.4.20. getFileInfo**

BSMD\_ERR getFileInfo ()

Input:

BSMD_HANDLE handle	本ライブラリの有効なハンドル
int *format	MIDI ファイルの SMF format
unsigned short *division	MIDI ファイルの分解能[TPQN]
unsigned long *totaltick	MIDI ファイルの総 tick 数
unsigned long *totaltime	MIDI ファイルの総演奏時間[s]

Output:

エラーコード

設定された MIDI ファイルの情報を取得する。

**CONFIDENTIAL**

#### 4.4.21. startFilePlay

BSMD\_ERR startFilePlay ()

Input:

BSMD\_HANDLE handle      本ライブラリの有効なハンドル

Output:

エラーコード

MIDI ファイルの演奏を現在の演奏位置から開始する。Real-time MIDI 再生中にのみ使用すること。

#### 4.4.22. stopFilePlay

BSMD\_ERR stopFilePlay ()

Input:

BSMD\_HANDLE handle      本ライブラリの有効なハンドル

Output:

エラーコード

現在行なわれている MIDI ファイルの演奏を停止(途中終了)する。

本関数は終了処理の開始を指示するものであり、本関数終了時点での終了処理の完了は保証されない。実際の演奏終了処理の完了は、後述するコールバックによる通知が行なわれた時点となる。

本関数による演奏停止後も演奏位置は保持される。

#### 4.4.23. seekFilePlay

BSMD\_ERR seekFilePlay ()

Input:

BSMD\_HANDLE handle      本ライブラリの有効なハンドル

unsigned long tick      演奏位置 [MIDI tick]

Output:

エラーコード

MIDI ファイルの演奏位置を設定する。

#### **4.4.24. isFilePlaying**

```
int isFilePlaying ()
```

Input:

BSMD\_HANDLE handle      本ライブラリの有効なハンドル

Output:

1: 演奏中

0: 非演奏中

MIDI ファイルの演奏中であるかを取得する。



#### **4.4.25. ctrl**

BSMD\_ERR ctrl ()

Input:

BSMD_HANDLE handle	本ライブラリの有効なハンドル
BSMD_CTRL ctrl	制御対象
void *data	制御データのアドレス
int size	制御データのサイズ[byte]

Output:

エラーコード

各種制御値の設定、取得を行う。制御対象（ctrl）毎に下記に詳細を述べる。

ctrl	data		内容
	type	I/O	
BSMD_CTRL_SET_SAMPLE_RATE	int	I	再生サンプリング周波数を設定する。 3.4.10 open により wave 出力デバイスが open されていない状態でのみ設定可能。
BSMD_CTRL_GET_SAMPLE_RATE	int	O	再生サンプリング周波数[Hz]を取得する。
BSMD_CTRL_SET_CHANNELS	int	I	未サポート
BSMD_CTRL_GET_CHANNELS	int	O	再生チャンネル数を取得する
BSMD_CTRL_SET_BLOCK_SIZE	long	I	再生処理フレームサイズ[sample]を設定する。3.4.10 open により wave 出力デバイスが open されていない状態でのみ設定可能。  Real-time MIDI 再生の時間分解能は本フレームサイズに依存する。 また、Real-time MIDI 再生のレイテンシ時間は本設定値および下記フレーム数設定に依存する。  ASIO/AudioUnit ドライバでは各デバイスの設定値が優先され本設定は有効化されない。したがってフレームサイズに基づく処理をアプリケーションで行う場合には、3.4.10 open 実行後に BSMD_CTRL_GET_BLOCK_SIZE にて実際の設定サイズを取得し利用すること。
BSMD_CTRL_GET_BLOCK_SIZE	long	O	再生処理フレームサイズ[sample]を取得する。
BSMD_CTRL_SET_BUFFERS	int	I	再生処理フレーム数を設定する。  Real-time MIDI 再生のレイテンシ時間は上記フレームサイズ設定および本設定値に依存する。  ASIO/AudioUnit ドライバでは本設定値は無視される(フレーム数 1 として考慮のこと)
BSMD_CTRL_GET_BUFFERS	int	O	再生処理フレーム数を取得する。
BSMD_CTRL_SET_POLY	int	I	シンセサイザの最大同時発音数を設定する。演奏中の変更は非対応とする。
BSMD_CTRL_GET_POLY	int	O	シンセサイザの最大同時発音数を取得する。

ctrl	data		内容
	type	I/O	
BSMD_CTRL_SET_MASTER_VOLUME	int	I	再生音量 (BSMD_VOLUME_MIN～BSMD_VOLUME_MAX)を設定する。初期設定値は BSMD_VOLUME_DEF とする。
BSMD_CTRL_GET_MASTER_VOLUME	int	O	再生音量を取得する。
BSMD_CTRL_SET_MASTER_KEY	int	I	キー (BSMD_KEY_MIN～BSMD_KEY_MAX)を設定する。動作は 100[cent]単位とし、初期設定値は BSMD_KEY_DEF とする。演奏開始および終了前後においても設定値は保持されるため、リセット処理はアプリケーション側で実装すること。
BSMD_CTRL_GET_MASTER_KEY	int	O	キーを取得する。
BSMD_CTRL_SET_MASTER_TUNE	int	I	ファインチューニング (BSMD_TUNE_MIN～BSMD_TUNE_MAX)を設定する。動作は 1[cent]単位とし初期設定値は BSMD_TUNE_DEF とする。演奏開始および終了前後においても設定値は保持されるため、リセット処理はアプリケーション側で実装すること。
BSMD_CTRL_GET_MASTER_TUNE	int	O	ファインチューニング値を取得する。
BSMD_CTRL_SET_SPEED	int	I	再生スピード (BSMD_SPEED_MIN～BSMD_SPEED_MAX)を設定する。動作は MIDI ファイルに設定されている演奏スピードの 1 [%]単位とし、初期設定値は BSMD_SPEED_DEF とする。演奏開始および終了前後においても設定値は保持されるため、リセット処理はアプリケーション側で実装すること。
BSMD_CTRL_GET_SPEED	int	O	再生スピードを取得する

ctrl	data		内容
	Type	I/O	
BSMD_CTRL_SET_REVERB	int	I	シンセサイザエンジンライブラリのリバーブ効果の有無を設定する。演奏開始および終了前後においても設定値は保持される。
BSMD_CTRL_GET_REVERB	int	O	シンセサイザエンジンライブラリのリバーブ効果の有無を取得する。
BSMD_CTRL_GET_REVERB_AVAILABLE	int	O	シンセサイザエンジンライブラリがリバーブ効果に対応しているかを取得する。
BSMD_CTRL_SET_CHORUS	int	I	シンセサイザエンジンライブラリのコーラス効果の有無を設定する。演奏開始および終了前後においても設定値は保持される。
BSMD_CTRL_GET_CHORUS	int	O	シンセサイザエンジンライブラリのコーラス効果の有無を取得する。
BSMD_CTRL_GET_CHORUS_AVAILABLE	int	O	シンセサイザエンジンライブラリがコーラス効果に対応しているかを取得する。
BSMD_CTRL_SET_DELAY	int	I	シンセサイザエンジンライブラリのディレイ効果の有無を設定する。演奏開始および終了前後においても設定値は保持される。
BSMD_CTRL_GET_DELAY	int	O	シンセサイザエンジンライブラリのディレイ効果の有無を取得する。
BSMD_CTRL_GET_DELAY_AVAILABLE	int	O	シンセサイザエンジンライブラリがディレイ効果に対応しているかを得る。
BSMD_CTRL_SET_REVERB_HQ	int	I	HQ リバーブを有効にする(1: On, 0: Off、カスタマイズバージョンのみ)

ctrl	data		内容
	type	I/O	
BSMD_CTRL_GET_SOUND_LIBRARY_NUM	int	O	音色ライブラリの最大格納数を取得する
BSMD_CTRL_SET_SOUND_LIBRARY	BSMD_SOUND_LIBRARY	I	ファイルより音色ライブラリを設定する
BSMD_CTRL_SET_SOUND_LIBRARY_MEMORY	BSMD_SOUND_LIBRARY_MEMORY	I	メモリより音色ライブラリを設定する
BSMD_CTRL_SET_SOUND_LIBRARY_SEL	BSMD_SOUND_LIBRARY_SEL	I	音色ライブラリの参照先を設定する
BSMD_CTRL_GET_SOUND_LIBRARY_SEL	BSMD_SOUND_LIBRARY_SEL	I/O	音色ライブラリの参照内容を取得する
BSMD_CTRL_SET_NUMBER_OF_REGIONS	int	I	音色あたりの最大 Region 数を設定する。

ctrl	data		内容
	type	I/O	
BSMD_CTRL_GET_INSTRUMENT_NAME ~ BSMD_CTRL_GET_INSTRUMENT_NAME + 15	char (TCHAR)	O	指定パート(Ch1~16)の現在の音色名を取得する
BSMD_CTRL_SET_MUTE ~ BSMD_CTRL_SET_MUTE + 15	int	I	指定パート(Ch1~16)のミュート設定を行う (0: Off, 1: On)
BSMD_CTRL_GET_MUTE ~ BSMD_CTRL_GET_MUTE + 15	int	O	指定パート(Ch1~16)のミュート設定を取得する (0: Off, 1: On)
BSMD_CTRL_SET_SOLO ~ BSMD_CTRL_SET_SOLO + 15	int	I	指定パート(Ch1~16)のソロ設定を行う (0: Off, 1: On)
BSMD_CTRL_GET_SOLO ~ BSMD_CTRL_GET_SOLO + 15	int	O	指定パート(Ch1~16)のソロ設定を取得する (0: Off, 1: On)

ctrl	data		内容
	type	I/O	
BSMD_CTRL_GET_AU DIO_UNIT			AudioUnitを取得する

#### **4.4.26. version**

```
void version ()
```

Input:

BSMD_HANDLE handle	本ライブラリの有効なハンドル
LPTSTR engine	シンセサイザエンジンライブラリのバージョン文字列
int engineSize	シンセサイザエンジンライブラリのバージョン文字数
LPTSTR driver	本ライブラリバージョン文字列
int driverSize	本ライブラリバージョン文字数

Output:

```
void
```

本ライブラリならびにシンセサイザエンジンライブラリのバージョン情報を得る。



#### **4.5. コールバック (BSMD\_CALLBACK)**

アプリケーションへ演奏中の各種情報を提供する。コールバック関数は下記の形式とし、コールバック用のユーザーパラメータと共に、4.4.1 initialize によって設定する。

各コールバックはシンセサイザの演算スレッドから呼び出される。従って、シンセサイザの動作へ影響を及ぼすことがないよう、各コールバック時のアプリケーション側の処理は必要最小限とし、処理時間を要するものに関してはコールバックはトリガとしての使用に留め、実処理自体は(カラオケ歌詞文字演算/表示処理等)は別スレッドにて行うよう配慮すること。

##### **4.5.1. Open**

type = BSMD\_CALLBACK\_TYPE\_OPEN, data = 未使用

Wave 出力ドライバがオープンされたことを示す。

##### **4.5.2. Close**

type = BSMD\_CALLBACK\_TYPE\_CLOSE, data = 未使用

Wave 出力ドライバがクローズされたことを示す。

##### **4.5.3. Start**

type = BSMD\_CALLBACK\_TYPE\_START, data = 未使用

Real-time MIDI 再生機能の開始を示す。

##### **4.5.4. Stop**

type = BSMD\_CALLBACK\_TYPE\_STOP, data = 未使用

Real-time MIDI 再生機能の停止を示す。

##### **4.5.5. Audio Frame**

type = BSMD\_CALLBACK\_TYPE\_FRAME, data = (BSMD\_FRAME \*) frameData

Wave 出力ドライバのフレーム処理が行われたことを通知する。

#### **4.5.6. File Start**

type = BSMD\_CALLBACK\_TYPE\_FILE\_START, data = 未使用

MIDI ファイル演奏の開始を示す。

#### **4.5.7. File Stop**

type = BSMD\_CALLBACK\_TYPE\_FILE\_STOP, data = (unsigned long \*) errorcode

MIDI ファイル演奏の終了処理完了を示す。本コールバックが呼ばれたスレッドにて次曲の演奏開始処理を行ってはならない。

errorcode は以下の演奏終了状態を示す。

0 : 演奏正常終了

BSMD\_ERR\_AUDIO\_DRIVER : Wave 出力ドライバにおける演奏異常終了

BSMD\_ERR\_DATA : データ異常による演奏異常終了

#### **4.5.8. File Seek**

type = BSMP\_CALLBACK\_TYPE\_FILE\_SEEK, data = 未使用

演奏位置が変更されたことを示す。

アプリケーションにおいて 4.5.9 MIDI クロック を使用して演奏位置算出を行っている場合には、本コールバック受信時に演奏位置を先頭に、テンポ情報を 120[BPM]にリセットすること。

#### **4.5.9. MIDI クロック**

type = BSMP\_CALLBACK\_TYPE\_CLOCK, data = 未使用

MIDI ファイルの演奏における MIDI クロックに相当するタイミング更新を示す。

#### **4.5.10. Tempo**

type = BSMP\_CALLBACK\_TYPE\_TEMPO, data = (unsigned long \*) tempo

MIDI ファイルの演奏におけるテンポ設定[usec/beat]の変更を示す。

#### **4.5.11. Time Signature**

type = BSMP\_CALLBACK\_TYPE\_TIME\_SIGNATURE, data = (unsigned long \*) timeSignature

MIDI ファイルの演奏における拍子設定(nn/dd/cc/bb)の変更を示す。

#### **4.5.12. Channel Message**

type = BSMP\_CALLBACK\_TYPE\_CHANNEL\_MESSAGE, data = (unsigned long \*) data

MIDI ファイルの演奏における Channel Message 送信を示す。

bit 31-24: MIDI Port (0x00 ~ )

bit 23 - 16: Status Byte (0x90 ~ 0xEF)

bit 15 - 8 : First Data (0x00 ~ 0x7F)

bit 7 - 0 : Second Data (0x00 ~ 0x7F)

#### **4.5.13. System Exclusive Message**

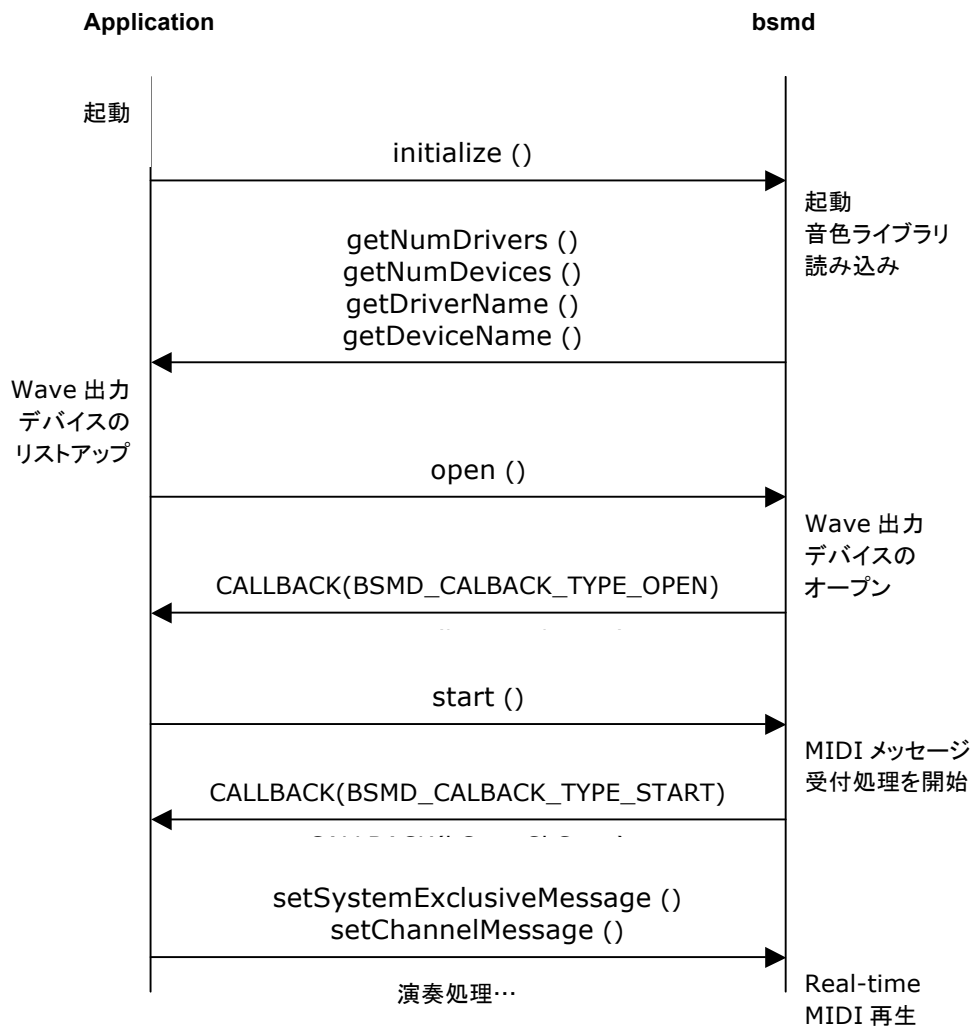
type = BSMP\_CALLBACK\_TYPE\_SYSTEM\_EXCLUSIVE\_MESSAGE, data = 未使用

MIDI ファイルの演奏における System Exclusive Message 送信を示す。

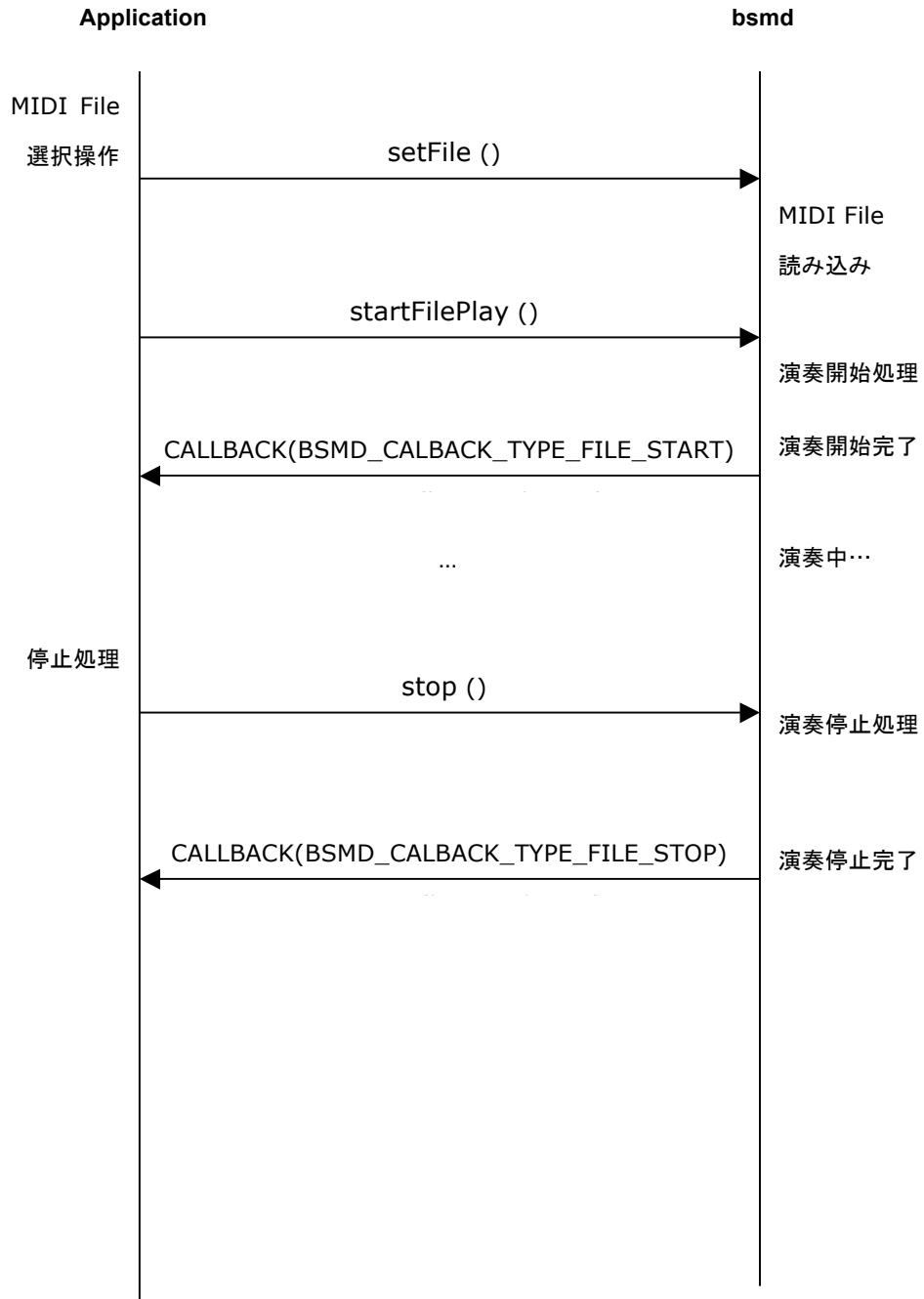
**CONFIDENTIAL**

## 4.6. 制御シーケンス

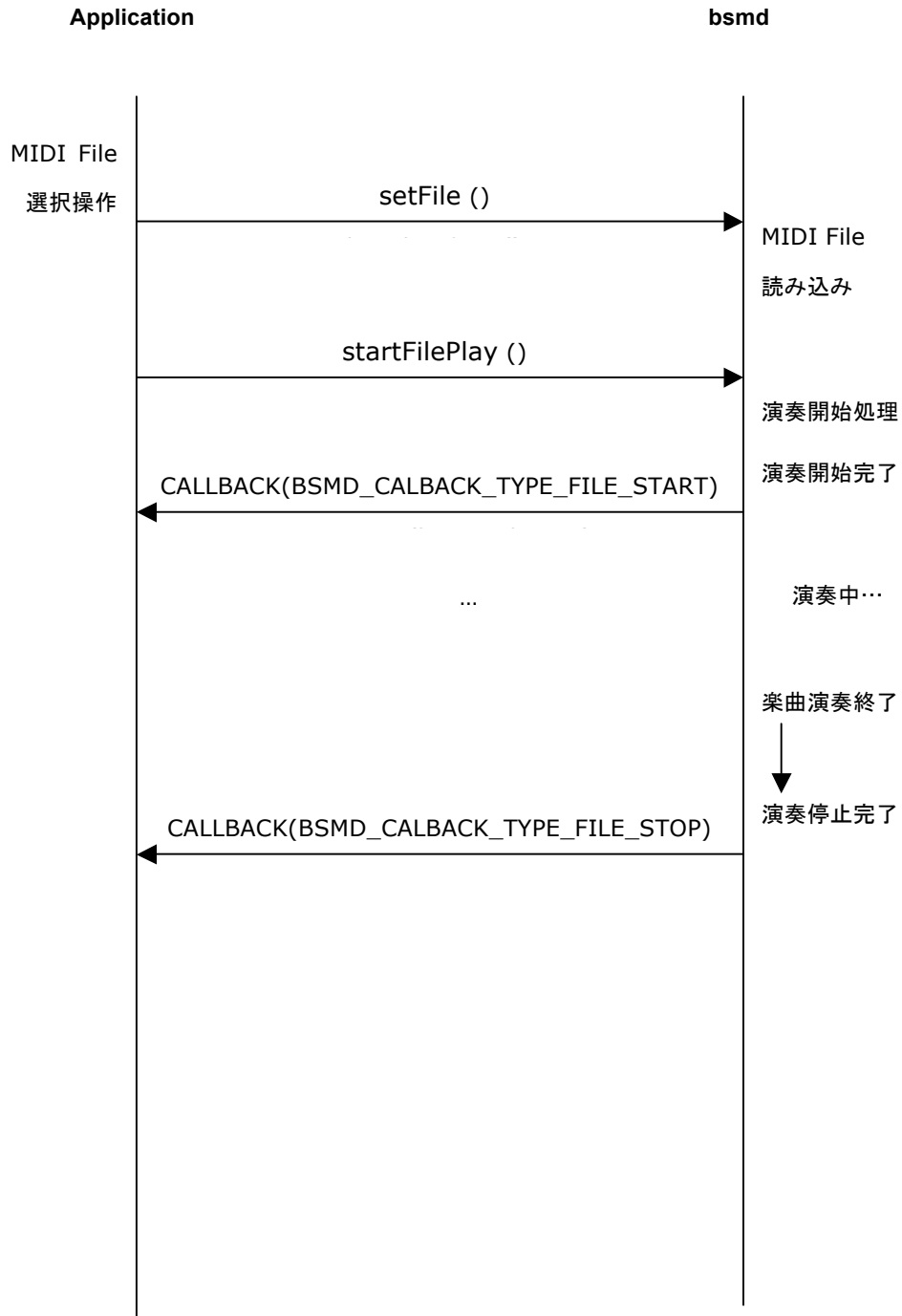
### 4.6.1. 起動および Real-time MIDI 再生の開始



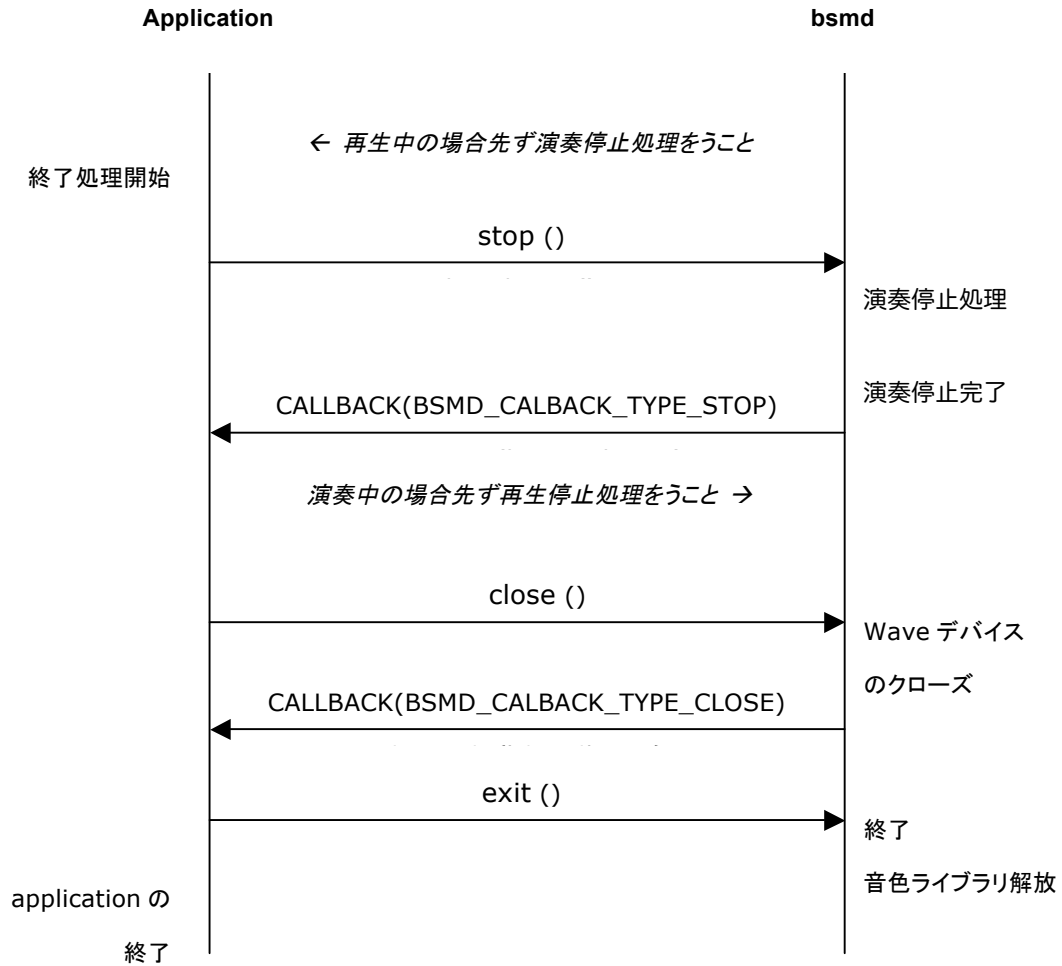
#### 4.6.2. MIDI ファイル指定 - 演奏開始 - 演奏終了



#### 4.6.3. MIDI ファイル指定 - 演奏開始 - 演奏自然終了



#### 4.6.4. ライブラリ終了



Real-time MIDI 再生中に終了処理を行う場合には先ず再生停止処理を行ってから Wave 出力デバイスのクローズ、およびライブラリの終了を行う。exit () を実行以後は登録されているコールバック関数はすべて呼び出されない。

## **5. 付録**

### **5.1. DLS フォーマットの対応について**

<wave-list>チャンクに含まれる Wave データは非圧縮 PCM、モノラル形式にのみ対応しています。

以下の Modulation Routing は対応されていません。デフォルト値での固定動作となります。

- Key Number Generator
  - MIDI Note to Key
  - RPN2 to Key
- Filter
  - Mod LFO CC1 to Fc
  - Mod LFO Channel Press. to Fc
- Gain
  - Mod LFO CC1 to Gain
  - Mod LFO Chan. Press. to Gain
  - Velocity to Gain
  - MIDI CC7 to Gain
  - MIDI CC11 to Gain
- Pitch
  - Pitch Wheel RPN0 to Pitch
  - RPN1 to Pitch
  - Vib LFO CC1 to Pitch
  - Vib LFO Chan. Press. to Pitch
  - Mod LFO CC1 to Pitch
  - Mod LFO Chan. Press. to Pitch
- Output
  - MIDI CC10 to Pan
  - Default Reverb Send
  - Default Chorus Send



**CONFIDENTIAL**

**Software Synthesizer  
MIDI Player / Driver Library  
Specification  
Version 3.0  
bismark.jp**