libvoice Reference

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Table of Contents

Datatypes

SceVoiceAppType

Enumeration for application type.

Definition

```
#include <voice_types.h>
typedef enum SceVoiceAppType {
    SCEVOICE_APPTYPE_GAME = 1 << 29
} SceVoiceAppType;
```

Enumeration Values

Macro	Value	Description
SCEVOICE_APPTYPE_GAME	1 << 29	Game Application.

Description

Enumeration for application type.

SceVoiceBasePortInfo

Base port information.

Definition

```
#include <voice_types.h>
typedef struct SceVoiceBasePortInfo {
    SceVoicePortType portType;
    SceVoicePortState state;
    uint32_t *pEdge;
    uint32_t numByte;
    uint32_t frameSize;
    uint16_t numEdge;
    uint16_t reserved;
} SceVoiceBasePortInfo;
```

Members

portType	Port type.
state	Port state.
pEdge	Connected ports.
numByte	When *portType* is an OPort: The number of bytes that can be removed from the buffer. When *portType* is an IPort: The number of bytes that can be inserted to the buffer.
frameSize	Unit frame size.
numEdge	Number of connections.
reserved	Reserved.

Description

Base port information. This structure holds the topology for the port returned from sceVoiceGetPortInfo().

See Also

sceVoiceGetPortInfo

SceVoiceBitRate

Enumeration for the bit rate for a port.

Definition

```
#include <voice_types.h>
typedef enum SceVoiceBitRate {
    SCEVOICE_BITRATE_NULL = -1,
    SCEVOICE_BITRATE_3850 = 3850,
    SCEVOICE_BITRATE_4650 = 4650,
    SCEVOICE_BITRATE_5700 = 5700,
    SCEVOICE_BITRATE_7300 = 7300
} SceVoiceBitRate;
```

Enumeration Values

Macro	Value	Description
`SCEVOICE_BITRATE_NULL`	`-1`	Null bitrate.
`SCEVOICE_BITRATE_3850`	`3850`	Bit rate of 3850 bps.
`SCEVOICE_BITRATE_4650`	`4650`	Bit rate of 4650 bps.
`SCEVOICE_BITRATE_5700`	`5700`	Bit rate of 5700 bps.
`SCEVOICE_BITRATE_7300`	`7300`	Bit rate of 7300 bps.

Description

Enumeration for the bit rate for a port.

See Also

`SceVoicePortParam`, `sceVoiceSetBitRate`

SceVoiceEventData

Data structure for voice event data.

Definition

```
#include <voice_types.h>
typedef struct SceVoiceEventData {
    SceVoiceEventType eventType;
    void *pUserData;
    union {
        struct {
            int16_t num;
            uint16_t pid[SCE_VOICE_MAX_OUT_VOICE_PORT];
        } dataReady;
        struct {
            bool bActive;
        } audioInput;
    };
} SceVoiceEventData;
```

Members

eventType Event type for this event data.

pUserData Arbitrary user data to pass to *onEvent* callback function.

num The number of SCEVOICE_PORTTYPE_OUT_VOICE ports with data ready.

pid The array of SCEVOICE_PORTTYPE_OUT_VOICE port ID.

dataReady Voice Data Ready structure.

bActive A flag that determines whether SCEVOICE_PORTTYPE_IN_DEVICE is muted by the AudioInput driver.

audioInput AudioInput status structure.

Description

Data structure for voice event data.

See Also

SceVoiceEventCallback

SceVoiceEventType

Enumeration for libvoice port event types.

Definition

```
#include <voice_types.h>
typedef enum SceVoiceEventType {
    SCEVOICE_EVENT_DATA_READY = 1 << 0,
    SCEVOICE_EVENT_AUDIOINPUT_STATUS = 1 << 1
} SceVoiceEventType;
```

Enumeration Values

Macro	Value	Description
SCEVOICE_EVENT_DATA_READY	1 << 0	Data is available only for the SCEVOICE_PORTTYPE_OUT_VOICE port.
SCEVOICE_EVENT_AUDIOINPUT_STATUS	1 << 1	AudioInput status, only for the SCEVOICE_PORTTYPE_IN_DEVICE port.

Description

Enumeration for libvoice port event types.

SceVoiceInitParam

Data structure for initializing libvoice.

Definition

```
#include <voice_types.h>
typedef struct SceVoiceInitParam {
    int32_t appType;
    SceVoiceEventCallback onEvent;
    void *pUserData;
    uint8_t reserved[32-sizeof(int32_t)*3];
} SceVoiceInitParam;
```

Members

appType Application configuration for Voice Service.

onEvent Application registers callback for receiving Voice Events. Pass `NULL` to disable the Voice Event service. Pass a callback function pointer to enable the Voice Event service.

pUserData Arbitrary user data to pass to *onEvent* callback function.

reserved Reserved bytes.

Description

Data structure for initializing libvoice.

See Also

sceVoiceInit

SceVoicePcmDataType

Enumeration for the PCM data type.

Definition

```
#include <voice_types.h>
typedef enum SceVoicePcmDataType {
    SCEVOICE_PCM_NULL = -1,
    SCEVOICE_PCM_SHORT_LITTLE_ENDIAN = 0
} SceVoicePcmDataType;
```

Enumeration Values

Macro	Value	Description
SCEVOICE_PCM_NULL	-1	Null datatype.
SCEVOICE_PCM_SHORT_LITTLE_ENDIAN	0	16-bit Short stored in little-endian format.

Description

Enumeration for the PCM data type.

See Also

SceVoicePCMFormat

SceVoicePCMFormat

Data structure for the PCM format.

Definition

```
#include <voice_types.h>
typedef struct SceVoicePCMFormat {
   SceVoicePcmDataType dataType;
   SceVoiceSamplingRate sampleRate;
} SceVoicePCMFormat;
```

Members

dataType	PCM data type.
sampleRate	Sampling rate.

Description

Data structure for the PCM format.

See Also

SceVoicePortParam

SceVoicePortAttr

Enumeration for port attributes.

Definition

```
#include <voice_types.h>
typedef enum SceVoicePortAttr {
    SCEVOICE_ATTR_AUDIOINPUT_ACTIVE = 1000
} SceVoicePortAttr;
```

Enumeration Values

Macro	Value	Description
SCEVOICE_ATTR_AUDIOINPUT_ACTIVE	1000	Attribute used to get the active state of an audio input device. This attribute only applies to SCEVOICE_PORTTYPE_IN_DEVICE. sceVoiceGetPortAttr() returns true when the audio input is active, and it returns false when the audio input port is inactive. The sceVoiceSetPortAttr() function is not applicable for this attribute.

Description

Enumeration for port attributes. Use sceVoiceGetPortAttr() to request user-set values for the attributes.

See Also

sceVoiceGetPortAttr, sceVoiceSetPortAttr

SceVoicePortParam

Port parameters.

Definition

```
#include <voice_types.h>
typedef struct SceVoicePortParam {
    SceVoicePortType portType;
    uint16_t threshold;
    uint16_t bMute;
    float volume;
    union {
        struct {
            SceVoiceBitRate bitrate;
        } voice;
        struct {
            uint32_t bufSize;
            SceVoicePCMFormat format;
        } pcmaudio;
        struct {
            uint32_t playerId;
        } device;
    };
} SceVoicePortParam;
```

Members

portType Port type.

threshold Shared buffer threshold (in millisecond). *threshold* is not applicable to a SCEVOICE_PORTTYPE_IN_DEVICE port or output port. A rough approximate maximum threshold value for IPort:

For SCEVOICE_PORTTYPE_IN_VOICE

- Bitrate 3850: 4000 ms
- Bitrate 4560: 3300 ms
- Bitrate 5700: 2600 ms
- Bitrate 7300: 2000 ms

For SCEVOICE_PORTTYPE_IN_PCMAUDIO

- 16-bit integer: (PCM-buffer-size/512*16) ms

bMute Indicates whether the port is muted by the application via sceVoiceSetMuteFlag().

volume Provides the port volume, a value of 0 silences the port.

bitrate Bit rate for the port.

voice Voice information data structure.

bufSize Buffer size.

format PCM format.

pcmaudio PCM audio information data structure.

playerId Player ID, set as 0 for now.

device Device information data structure.

Description

Port parameters. This structure is used to create and update ports.

See Also

sceVoiceCreatePort, sceVoiceUpdatePort

SceVoicePortState

Enumeration for the state of a port.

Definition

```
#include <voice_types.h>
typedef enum SceVoicePortState {
    SCEVOICE_PORTSTATE_NULL = -1,
    SCEVOICE_PORTSTATE_IDLE = 0,
    SCEVOICE_PORTSTATE_READY = 1,
    SCEVOICE_PORTSTATE_BUFFERING = 2,
    SCEVOICE_PORTSTATE_RUNNING = 3
} SceVoicePortState;
```

Enumeration Values

Macro	Value	Description
SCEVOICE_PORTSTATE_NULL	-1	The initial port state.
SCEVOICE_PORTSTATE_IDLE	0	The port is idle.
SCEVOICE_PORTSTATE_READY	1	The port is ready.
SCEVOICE_PORTSTATE_BUFFERING	2	The port is buffering data.
SCEVOICE_PORTSTATE_RUNNING	3	The port is running.

Description

Enumeration for the state of a port. Each port has its own state. State changes can be the result of events generated by Voice Service, of libvoice function calls made directly by the application, or of the data level of the buffer.

The following diagram shows how the states change with specific actions or events on the port:

```
NULL ---------------- Topology Build Completed ----------------> IDLE
IDLE -------------------- sceVoiceStart ------------------------> READY
READY ------------ SCEVOICE_EVENT_SERVICE_ATTACHED ------------> BUFFERING
BUFFERING ------------ Data reaches threshold -----------------> RUNNING
RUNNING ---- SCEVOICE_EVENT_SERVICE_DETACHED/sceVoiceStop -----> IDLE
READY ------------------- sceVoiceStop -------------------------> IDLE
```

See Also

SceVoiceBasePortInfo

SceVoicePortType

Port types for libvoice.

Definition

```
#include <voice_types.h>
typedef enum SceVoicePortType {
    SCEVOICE_PORTTYPE_NULL = -1,
    SCEVOICE_PORTTYPE_IN_DEVICE = 0,
    SCEVOICE_PORTTYPE_IN_PCMAUDIO = 1,
    SCEVOICE_PORTTYPE_IN_VOICE = 2,
    SCEVOICE_PORTTYPE_OUT_PCMAUDIO = 3,
    SCEVOICE_PORTTYPE_OUT_VOICE = 4,
    SCEVOICE_PORTTYPE_OUT_DEVICE = 5
} SceVoicePortType;
```

Enumeration Values

Macro	Value	Description
SCEVOICE_PORTTYPE_NULL	-1	Null port.
SCEVOICE_PORTTYPE_IN_DEVICE	0	Audio input stream port.
SCEVOICE_PORTTYPE_IN_PCMAUDIO	1	PCM input stream port.
SCEVOICE_PORTTYPE_IN_VOICE	2	Voice input stream port.
SCEVOICE_PORTTYPE_OUT_PCMAUDIO	3	PCM output stream port.
SCEVOICE_PORTTYPE_OUT_VOICE	4	Voice output stream port.
SCEVOICE_PORTTYPE_OUT_DEVICE	5	Audio output stream port.

Description

Port types for libvoice.

See Also

SceVoiceBasePortInfo, SceVoicePortParam

SceVoiceResourceInfo

Data structure for maximum voice resource support.

Definition

```
#include <voice_types.h>
typedef struct SceVoiceResourceInfo {
   uint16_t maxInVoicePort;
   uint16_t maxOutVoicePort;
   uint16_t maxInDevicePort;
   uint16_t maxOutDevicePort;
   uint16_t maxTotalPort;
} SceVoiceResourceInfo;
```

Members

maxInVoicePort	Maximum number of SCEVOICE_PORTTYPE_IN_VOICE ports supported.
maxOutVoicePort	Maximum number of SCEVOICE_PORTTYPE_OUT_VOICE ports supported.
maxInDevicePort	Maximum number of SCEVOICE_PORTTYPE_IN_DEVICE ports supported.
maxOutDevicePort	Maximum number of SCEVOICE_PORTTYPE_OUT_DEVICE ports supported.
maxTotalPort	Maximum number ports supported.

Description

Data structure for maximum voice resource support.

See Also

sceVoiceGetResourceInfo

SceVoiceSamplingRate

Enumeration for the sampling rate for a port.

Definition

```
#include <voice_types.h>
typedef enum SceVoiceSamplingRate {
   SCEVOICE_SAMPLINGRATE_NULL = -1,
   SCEVOICE_SAMPLINGRATE_16000 = 16000
} SceVoiceSamplingRate;
```

Enumeration Values

Macro	Value	Description
SCEVOICE_SAMPLINGRATE_NULL	-1	Null sampling rate.
SCEVOICE_SAMPLINGRATE_16000	16000	16000 sampling rate

Description

Enumeration for the sampling rate for a port.

See Also

SceVoicePCMFormat, sceVoiceCreatePort

SceVoiceStartParam

Start parameters.

Definition

```
#include <voice_types.h>
typedef struct SceVoiceStartParam {
   int32_t container;
   uint8_t reserved[32-sizeof(int32_t)*1];
} SceVoiceStartParam;
```

Members

container	Memory address allocated by the application.
reserved	Reserved bytes.

Description

Start parameters. This structure is used to start the voice service.

See Also

sceVoiceStart

SceVoiceVersion

Enumeration for SceVoice structure version check; for future structure extension.

Definition

```
#include <voice_types.h>
typedef enum SceVoiceVersion {
   SCEVOICE_VERSION_100 = 100
} SceVoiceVersion;
```

Enumeration Values

Macro	Value	Description
SCEVOICE_VERSION_100	100	Version 100.

Description

Enumeration for SceVoice structure version check; for future structure extension.

Functions

sceVoiceCheckTopology

Checks the created port connections to see if their topology is valid for streaming.

Definition

```
#include <voice.h>
int sceVoiceCheckTopology(void);
```

Return Values

Value	Description
`SCE_OK`	Topology is valid.
`SCE_VOICE_ERROR_TOPOLOGY`	Topology error.

Description

Checks the created port connections to see if their topology is valid for streaming. This function is provided as a helper function for developers to verify the topology layout during development.

The following cases are currently being checked in a topology:

- Every port in the topology should be connected to one or more ports. No single unconnected port is allowed in the topology.
- When a `SCEVOICE_PORTTYPE_IN_DEVICE` port and a `SCEVOICE_PORTTYPE_IN_PCMAUDIO` port both attempt to encode the data, both ports should be connected to a same `SCEVOICE_PORTTYPE_OUT_VOICE` port.
- When several `SCEVOICE_PORTTYPE_OUT_VOICE` ports are created, all related voice out parameters (*`bMute`*, *`volume`*, *`bitrate`*) must be all the same because there is only one encoder available inside libvoice.

sceVoiceConnectIPortToOPort

Connects the specified input port (IPort) to the specified output port (OPort).

Definition

```
#include <voice.h>
int sceVoiceConnectIPortToOPort(
    uint32_t ips,
    uint32_t ops
);
```

Arguments

ips	[in] The input port ID.
ops	[in] The output port ID.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_VOICE_ERROR_LIBVOICE_NOT_INIT	The libvoice library has not been initialized.
SCE_VOICE_ERROR_TOPOLOGY	Topology error.
SCE_VOICE_ERROR_PORT_INVALID	Invalid port.
SCE_VOICE_ERROR_RESOURCE_INSUFFICIENT	Insufficient resources.

Description

Connects the specified input port (IPort) to the specified output port (OPort). To avoid recursive connections, the reverse connection is not allowed. That is, connecting OPort to IPort is not allowed.

See Also

sceVoiceDisconnectIPortFromOPort

sceVoiceCreatePort

Creates the inbound input port (IPort) and the outbound output port (OPort).

Definition

```
#include <voice.h>
int sceVoiceCreatePort(
    uint32_t *portId,
    const SceVoicePortParam *pArg
);
```

Arguments

portId	[out] The port ID.
pArg	[in] Port Parameters.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_VOICE_ERROR_LIBVOICE_NOT_INIT	The libvoice library has not been initialized.
SCE_VOICE_ERROR_ARGUMENT_INVALID	Invalid argument.
SCE_VOICE_ERROR_RESOURCE_INSUFFICIENT	Insufficient resources.
SCE_VOICE_ERROR_GENERAL	Error from non-voice functions.

Description

Creates the inbound input port (IPort) and the outbound output port (OPort).

sceVoiceDeletePort

Deletes the specified port.

Definition

```
#include <voice.h>
int sceVoiceDeletePort(
   uint32_t portId
);
```

Arguments

portId	[in] The port ID.

Return Values

Value	Description
`SCE_OK`	Successful completion.
`SCE_VOICE_ERROR_LIBVOICE_NOT_INIT`	The libvoice library has not been initialized.
`SCE_VOICE_ERROR_TOPOLOGY`	Topology error.
`SCE_VOICE_ERROR_PORT_INVALID`	Invalid port.
`SCE_VOICE_ERROR_GENERAL`	Error from non-voice functions.

Description

Deletes the specified port. It releases the port memory and binding. If the port is a microphone or a speaker device, it also deletes the device.

sceVoiceDisconnectIPortFromOPort

Disconnects the specified input port (IPort) from the specified output port (OPort).

Definition

```
#include <voice.h>
int sceVoiceDisconnectIPortFromOPort(
    uint32_t ips,
    uint32_t ops
);
```

Arguments

ips	[in] The input port ID.
ops	[in] The output port ID.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_VOICE_ERROR_LIBVOICE_NOT_INIT	The libvoice library has not been initialized.
SCE_VOICE_ERROR_TOPOLOGY	Topology error.
SCE_VOICE_ERROR_PORT_INVALID	Invalid port.

Description

Disconnects the specified input port (IPort) from the specified output port (OPort).

See Also

sceVoiceConnectIPortToOPort

sceVoiceEnd

Terminates Voice Service.

Definition

```
#include <voice.h>
int sceVoiceEnd(void);
```

Return Values

Value	Description
`SCE_OK`	Successful completion.
`SCE_VOICE_ERROR_LIBVOICE_NOT_INIT`	The libvoice library has not been initialized.
`SCE_VOICE_ERROR_TOPOLOGY`	Topology error.

Description

Terminates Voice Service. `sceVoiceEnd()` does the following:

- Frees resources.
- Exits Voice Service.

sceVoiceGetBitRate

Gets the actual bit-rate value for the specified port.

Definition

```
#include <voice.h>
int sceVoiceGetBitRate(
    uint32_t portId,
    uint32_t *bitrate
);
```

Arguments

portId	[in] The port ID.
bitrate	[out] The actual bit-rate value (not the enumeration).

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_VOICE_ERROR_LIBVOICE_NOT_INIT	The libvoice library has not been initialized.
SCE_VOICE_ERROR_TOPOLOGY	Topology error.
SCE_VOICE_ERROR_ARGUMENT_INVALID	Invalid argument.
SCE_VOICE_ERROR_PORT_INVALID	Invalid port.

Description

Gets the actual bit-rate value for the specified port.

sceVoiceGetMuteFlag

Gets the mute flag for the specified port.

Definition

```
#include <voice.h>
int sceVoiceGetMuteFlag(
    uint32_t portId,
    uint16_t *bMuted
);
```

Arguments

portId	[in] The port ID.
bMuted	[out] The address containing the *bMute* flag value.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_VOICE_ERROR_LIBVOICE_NOT_INIT	The libvoice library has not been initialized.
SCE_VOICE_ERROR_TOPOLOGY	Topology error.
SCE_VOICE_ERROR_ARGUMENT_INVALID	Invalid argument.
SCE_VOICE_ERROR_PORT_INVALID	Invalid port.

Description

Gets the mute flag for the specified port.

See Also

sceVoiceSetMuteFlag, sceVoiceSetMuteFlagAll

sceVoiceGetPortAttr

Gets the user-set values of the specified port attribute for the specified port.

Definition

```
#include <voice.h>
int sceVoiceGetPortAttr(
    uint32_t portId,
    SceVoicePortAttr attr,
    void *attrValue
);
```

Arguments

portId	[in] The port ID.
attr	[in] An enumeration value that specifies the port attribute for which to get the value.
attrValue	[out] The address of the actual user-set value.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_VOICE_ERROR_LIBVOICE_NOT_INIT	The libvoice library has not been initialized.
SCE_VOICE_ERROR_ARGUMENT_INVALID	Invalid argument.
SCE_VOICE_ERROR_PORT_INVALID	Invalid port.
SCE_VOICE_ERROR_TOPOLOGY	Topology error.

Description

Gets the user-set values of the specified port attribute for the specified port.

sceVoiceGetPortInfo

Gets basic port information for the specified port.

Definition

```
#include <voice.h>
int sceVoiceGetPortInfo(
    uint32_t portId,
    SceVoiceBasePortInfo *pInfo
);
```

Arguments

portId	[in] The port ID.
pInfo	[in/out] The pointer to the basic port information. The application must allocate the memory for *pInfo->pEdge* if it wants the array of the port's connection edge information returned. Voice Service goes through the internal binding to get the edge connection information. This is a time-consuming process. The application should pass *pInfo->pEdge* as `NULL` if it does not need to know the connection edges.

Return Values

Value	Description
`SCE_OK`	Successful completion.
`SCE_VOICE_ERROR_LIBVOICE_NOT_INIT`	The libvoice library has not been initialized.
`SCE_VOICE_ERROR_TOPOLOGY`	Topology error.
`SCE_VOICE_ERROR_ARGUMENT_INVALID`	Invalid argument.
`SCE_VOICE_ERROR_PORT_INVALID`	Invalid port.
`SCE_VOICE_ERROR_SERVICE_DETACHED`	System Voice Service is detached.

Description

Gets basic port information for the specified port.

sceVoiceGetResourceInfo

Get the resource information.

Definition

```
#include <voice.h>
int sceVoiceGetResourceInfo(
    SceVoiceResourceInfo *pInfo
);
```

Arguments

pInfo [in/out] The pointer to the voice resource information.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_VOICE_ERROR_LIBVOICE_NOT_INIT	The libvoice library has not been initialized.
SCE_VOICE_ERROR_ARGUMENT_INVALID	Invalid argument.

Description

Get the resource information.

sceVoiceGetVolume

Gets the volume for the specified port.

Definition

```
#include <voice.h>
int sceVoiceGetVolume(
    uint32_t portId,
    float *volume
);
```

Arguments

[in] *portId*	[in] The port ID.
[out] *volume*	[out] The address containing the value of the volume.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_VOICE_ERROR_LIBVOICE_NOT_INIT	The libvoice library has not been initialized.
SCE_VOICE_ERROR_TOPOLOGY	Topology error.
SCE_VOICE_ERROR_ARGUMENT_INVALID	Invalid argument.
SCE_VOICE_ERROR_PORT_INVALID	Invalid port.

Description

Gets the volume for the specified port.

See Also

sceVoiceSetVolume

sceVoiceInit

Initializes Voice Service.

Definition

```
#include <voice.h>
int sceVoiceInit(
    SceVoiceInitParam *pArg,
    SceVoiceVersion version
);
```

Arguments

pArg	[in] Initialization parameters.
version	[in] Version for future extension.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_VOICE_ERROR_LIBVOICE_INITIALIZED	The libvoice library has already been initialized.
SCE_VOICE_ERROR_RESOURCE_INSUFFICIENT	Insufficient resources.
SCE_VOICE_ERROR_ARGUMENT_INVALID	Invalid argument.
SCE_VOICE_ERROR_TOPOLOGY	Topology error.

Description

Initializes Voice Service. `sceVoiceInit()` does the following:

- Initializes the Voice Service.
- Allocates resources for the Voice Service.

There are two methods to get Voice System status: One is via a Voice Event, where the application registers a callback *onEvent* for receiving voice events. The other method is via polling, where the application calls `sceVoiceGetPortInfo()` to get the port status (see `SceVoicePortState`).

sceVoicePausePort

Pauses streaming of the specified port.

Definition

```
#include <voice.h>
int sceVoicePausePort(
   uint32_t portId
);
```

Arguments

portId	[in] The port ID.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_VOICE_ERROR_LIBVOICE_NOT_INIT	The libvoice library has not been initialized.
SCE_VOICE_ERROR_TOPOLOGY	Topology error.
SCE_VOICE_ERROR_PORT_INVALID	Invalid port.

Description

Pauses streaming of the specified port.

sceVoicePausePortAll

Pauses streaming for all ports.

Definition

```
#include <voice.h>
int sceVoicePausePortAll(void);
```

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_VOICE_ERROR_LIBVOICE_NOT_INIT	The libvoice library has not been initialized.
SCE_VOICE_ERROR_TOPOLOGY	Topology error.

Description

Pauses streaming for all ports.

sceVoiceReadFromOPort

Reads data from the specified output port's (OPort) output buffer.

Definition

```
#include <voice.h>
int sceVoiceReadFromOPort(
    uint32_t ops,
    void *data,
    uint32_t *size
);
```

Arguments

ops	[in] The output port ID.
data	[out] The buffer pointer to hold the read data.
size	[in/out] For input, the number of requested bytes. For output, the actual number of bytes read.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_VOICE_ERROR_LIBVOICE_NOT_INIT	The libvoice library has not been initialized.
SCE_VOICE_ERROR_TOPOLOGY	Topology error.
SCE_VOICE_ERROR_ARGUMENT_INVALID	Invalid argument.
SCE_VOICE_ERROR_PORT_INVALID	Invalid port.
SCE_VOICE_ERROR_SERVICE_DETACHED	System Voice Service is detached.

Description

Reads data from the specified output port's (OPort) output buffer.

sceVoiceResetPort

Flushes the port’s internal buffer.

Definition

```
#include <voice.h>
int sceVoiceResetPort(
   uint32_t portId
);
```

Arguments

portId	[in] The port ID.

Return Values

Value	Description
`SCE_OK`	Successful completion.
`SCE_VOICE_ERROR_LIBVOICE_NOT_INIT`	The libvoice library has not been initialized.
`SCE_VOICE_ERROR_TOPOLOGY`	Topology error.
`SCE_VOICE_ERROR_PORT_INVALID`	Invalid port.

Description

Flushes the port’s internal buffer.

sceVoiceResumePort

Resumes streaming of the specified port.

Definition

```
#include <voice.h>
int sceVoiceResumePort(
   uint32_t portId
);
```

Arguments

portId	[in] The port ID.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_VOICE_ERROR_LIBVOICE_NOT_INIT	The libvoice library has not been initialized.
SCE_VOICE_ERROR_TOPOLOGY	Topology error.
SCE_VOICE_ERROR_PORT_INVALID	Invalid port.

Description

Resumes streaming of the specified port.

sceVoiceResumePortAll

Resumes streaming for all ports.

Definition

```
#include <voice.h>
int sceVoiceResumePortAll(void);
```

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_VOICE_ERROR_LIBVOICE_NOT_INIT	The libvoice library has not been initialized.
SCE_VOICE_ERROR_TOPOLOGY	Topology error.

Description

Resumes streaming for all ports.

sceVoiceSetBitRate

Sets the specific bit rate for the specified port.

Definition

```
#include <voice.h>
int sceVoiceSetBitRate(
    uint32_t portId,
    SceVoiceBitRate bitrate
);
```

Arguments

portId	[in] The port ID.
bitrate	[in] The bit rate, provided by the enumeration.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_VOICE_ERROR_LIBVOICE_NOT_INIT	The libvoice library has not been initialized.
SCE_VOICE_ERROR_TOPOLOGY	Topology error.
SCE_VOICE_ERROR_ARGUMENT_INVALID	Invalid argument.
SCE_VOICE_ERROR_PORT_INVALID	Invalid port.

Description

Sets the specific bit rate for the specified port.

sceVoiceSetMuteFlag

Enables or disables mute for the specified port.

Definition

```
#include <voice.h>
int sceVoiceSetMuteFlag(
    uint32_t portId,
    uint16_t bMuted
);
```

Arguments

portId	[in] The port ID.
bMuted	[in] If true, mute is enabled for the port, no audio is generated. If false, mute is disabled for the port, audio is generated.

Return Values

Value	Description
`SCE_OK`	Successful completion.
`SCE_VOICE_ERROR_LIBVOICE_NOT_INIT`	The libvoice library has not been initialized.
`SCE_VOICE_ERROR_TOPOLOGY`	Topology error.
`SCE_VOICE_ERROR_PORT_INVALID`	Invalid port.

Description

Enables or disables mute for the specified port. When a port is muted, its data is still being processed but with an effective volume of 0.

See Also

`sceVoiceGetMuteFlag`, `sceVoiceSetMuteFlagAll`

sceVoiceSetMuteFlagAll

Enables or disables mute at all input ports (IPorts) and output ports (OPorts).

Definition

```
#include <voice.h>
int sceVoiceSetMuteFlagAll(
    uint16_t bMuted
);
```

Arguments

bMuted	[in] If true, enables mute for all the IPorts/OPorts, no audio is generated. If false, the audio generation depends on the `SceVoicePortParam.`*`bMute`* flag setting for each port.

Return Values

Value	Description
`SCE_OK`	Successful completion.
`SCE_VOICE_ERROR_LIBVOICE_NOT_INIT`	The libvoice library has not been initialized.
`SCE_VOICE_ERROR_TOPOLOGY`	Topology error.

Description

Enables or disables mute at all input ports (IPorts) and output ports (OPorts). This does not change the `SceVoicePortParam.`*`bMute`* flag setting for each port. When a port is muted, its data is still being processed but with an effective volume of 0.

See Also

`SceVoicePortParam`

sceVoiceSetPortAttr

Sets a port's attribute to a specified value.

Definition

```
#include <voice.h>
int sceVoiceSetPortAttr(
    uint32_t portId,
    SceVoicePortAttr attr,
    void *attrValue
);
```

Arguments

portId	[in] The port ID.
attr	[in] An enumeration value that specifies the port attribute for which to set a value.
attrValue	[in] The new attribute value for the port.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_VOICE_ERROR_LIBVOICE_NOT_INIT	The libvoice library has not been initialized.
SCE_VOICE_ERROR_ARGUMENT_INVALID	Invalid argument.
SCE_VOICE_ERROR_PORT_INVALID	Invalid port.
SCE_VOICE_ERROR_TOPOLOGY	Topology error.

Description

Sets a port's attribute to a specified value. See `SceVoicePortAttr()` for a description of the attributes for which the application can set values.

sceVoiceSetVolume

Sets the volume for the specified port.

Definition

```
#include <voice.h>
int sceVoiceSetVolume(
    uint32_t portId,
    float volume
);
```

Arguments

portId	[in] The port ID.
volume	[in] The value of volume. A value of 0 silences the port.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_VOICE_ERROR_LIBVOICE_NOT_INIT	The libvoice library has not been initialized.
SCE_VOICE_ERROR_TOPOLOGY	Topology error.
SCE_VOICE_ERROR_PORT_INVALID	Invalid port.

Description

Sets the volume for the specified port.

See Also

sceVoiceGetVolume

sceVoiceStart

Starts the Voice Service with user provided memory.

Definition

```
#include <voice.h>
int sceVoiceStart(
    SceVoiceStartParam *pArg
);
```

Arguments

pArg [in] System Voice Service start parameters.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_VOICE_ERROR_LIBVOICE_NOT_INIT	The libvoice library has not been initialized.
SCE_VOICE_ERROR_TOPOLOGY	Topology error.
SCE_VOICE_ERROR_CONTAINER_INVALID	Memory container error.
SCE_VOICE_ERROR_ARGUMENT_INVALID	Invalid argument.
SCE_VOICE_ERROR_GENERAL	Error from non-voice functions.

Description

Starts the Voice Service with user provided memory. The application provides a memory container and calls sceVoiceStart() to request the start of the Voice Service. Changes the port state from IDLE to READY if the topology validation is successful.

sceVoiceStop

Stops the Voice Service.

Definition

```
#include <voice.h>
int sceVoiceStop(void);
```

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_VOICE_ERROR_LIBVOICE_NOT_INIT	The libvoice library has not been initialized.
SCE_VOICE_ERROR_TOPOLOGY	Topology error.
SCE_VOICE_ERROR_GENERAL	Error from non-voice functions.

Description

Stops the Voice Service. The Voice Service releases memory and detaches.

sceVoiceUpdatePort

Updates the specified port with parameters specified to the SceVoicePortParam structure.

Definition

```
#include <voice.h>
int sceVoiceUpdatePort(
    uint32_t portId,
    const SceVoicePortParam *pArg
);
```

Arguments

portId	[in] The port ID.
pArg	[in] Port parameters.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_VOICE_ERROR_LIBVOICE_NOT_INIT	The libvoice library has not been initialized.
SCE_VOICE_ERROR_TOPOLOGY	Topology error.
SCE_VOICE_ERROR_ARGUMENT_INVALID	Invalid argument.
SCE_VOICE_ERROR_PORT_INVALID	Invalid port.
SCE_VOICE_ERROR_GENERAL	Error from non-voice functions.

Description

Updates the specified port with parameters specified to the SceVoicePortParam structure. *portType* cannot be updated. There is no effect in updating device *playerId* and pcmaudio *bufSize*.

sceVoiceWriteToIPort

Writes data to the specified input port's (IPort) input buffer.

Definition

```
#include <voice.h>
int sceVoiceWriteToIPort(
    uint32_t ips,
    const void *data,
    uint32_t *size,
    int16_t frameGaps
);
```

Arguments

ips	[in] The input port ID.
data	[in] A constant buffer pointer to hold the write data.
size	[in/out] For input, the size of the data to write. For output, the actual number of bytes written.
frameGaps	[in] The number of frames dropped between voice data writings. Values include:

- 0: For a continuous stream with no dropped frames.
- Positive value (+X): For X dropped forward frames. For example, if the current top frame index were 5, the next expected continuous frame index would be 6. If instead frame index 8 arrives, `frameGaps` is the difference between the received frame index (8) and the expected continuous frame index (6): 8 - 6 = 2. The next expected frame index is now 9.
- Negative value (-X): For X dropped preceding frames. For example, if the current top frame index were 5, the next expected continuous frame index would be 6. If instead frame index 2 arrives, `frameGaps` is the difference between the received frame index (2) and the expected frame index (6): 2 - 6 = -4. The next expected frame index is still 6.

Return Values

Value	Description
`SCE_OK`	Successful completion.
`SCE_VOICE_ERROR_LIBVOICE_NOT_INIT`	The libvoice library has not been initialized.
`SCE_VOICE_ERROR_TOPOLOGY`	Topology error.
`SCE_VOICE_ERROR_ARGUMENT_INVALID`	Invalid argument.
`SCE_VOICE_ERROR_PORT_INVALID`	Invalid port.
`SCE_VOICE_ERROR_SERVICE_DETACHED`	System Voice Service is detached.

Description

Writes data to the specified input port's (IPort) input buffer.

Callbacks

SceVoiceEventCallback

Voice Event Callback Function.

Definition

```
#include <voice_types.h>
typedef void (*SceVoiceEventCallback)(
   SceVoiceEventData *pEvent
);
```

Arguments

pEvent	See SceVoiceEventData.

Return Values

None

Description

Voice Event Callback Function.

Defines

Define Summary

Define	Value	Description
SCE_VOICE_ERROR_ARGUMENT_INVALID	SCE_ERROR_CAST (0x804e0805)	Invalid argument.
SCE_VOICE_ERROR_CONTAINER_INVALID	SCE_ERROR_CAST (0x804e0806)	Invalid container.
SCE_VOICE_ERROR_GENERAL	SCE_ERROR_CAST (0x804e0803)	Errors that belong to non-voice APIs.
SCE_VOICE_ERROR_LIBVOICE_INITIALIZED	SCE_ERROR_CAST (0x804e0802)	The libvoice library has already been initialized.
SCE_VOICE_ERROR_LIBVOICE_NOT_INIT	SCE_ERROR_CAST (0x804e0801)	The libvoice library has not been initialized.
SCE_VOICE_ERROR_NOT_IMPLEMENTED	SCE_ERROR_CAST (0x804e0809)	Feature not yet implemented.
SCE_VOICE_ERROR_PORT_INVALID	SCE_ERROR_CAST (0x804e0804)	Invalid port.
SCE_VOICE_ERROR_RESOURCE_INSUFFICIENT	SCE_ERROR_CAST (0x804e0808)	Insufficient resources.
SCE_VOICE_ERROR_SERVICE_ATTACHED	SCE_ERROR_CAST (0x804e080b)	System Voice Service is attached.
SCE_VOICE_ERROR_SERVICE_DETACHED	SCE_ERROR_CAST (0x804e080a)	System Voice Service is detached.
SCE_VOICE_ERROR_TOPOLOGY	SCE_ERROR_CAST (0x804e0807)	Topology error.
SCE_VOICE_INVALID_PORT_ID	0xff	Invalid port ID.
SCE_VOICE_MAX_OUT_VOICE_PORT	7	The maximum number of SCEVOICE_PORTTYPE_OUT_VOICE ports.
SCE_VOICE_MEMORY_CONTAINER_SIZE	0x40000	The minimum size of memory required by libvoice.