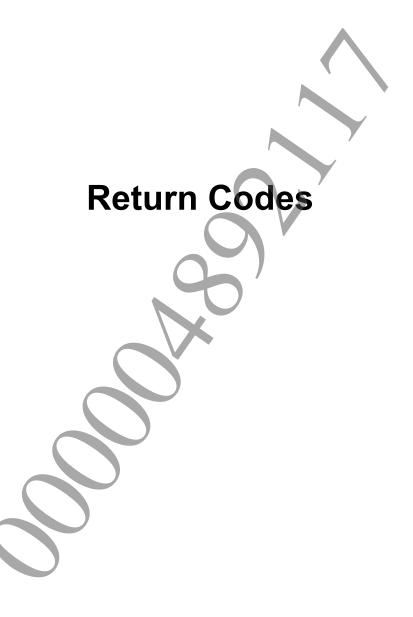


© 2012 Sony Computer Entertainment Inc. All Rights Reserved. SCE Confidential

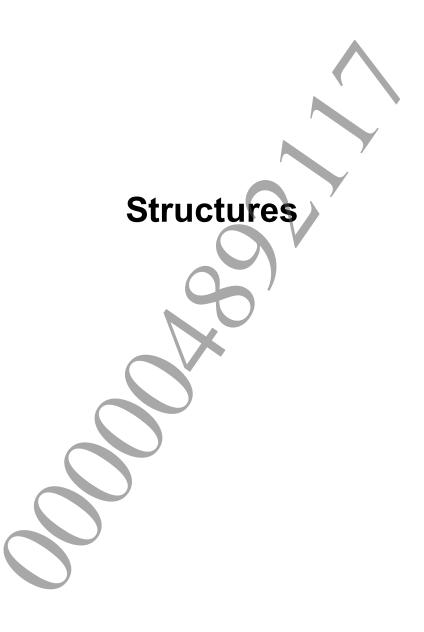
Table of Contents

Return Codes	
Return Codes	
Structures	
SceSulphaConfig	
Type Definitions	
SceSulphaUTF8	
SceSulphaCallback	
Functions	10
sceSulphaFileConnect	
sceSulphaFileDisconnect	
sceSulphaGetDefaultConfig	1;
sceSulphaGetNeededMemory	
sceSulphalnit	
sceSulphaNetworkInit	
sceSulphaNetworkShutdown	17
sceSulphaSetBookmark	18
sceSulphaShutdown	19
sceSulphaUpdate	20
NGS Structures	2 [,]
SceSulphaNgsConfig	
NGS Functions	23
sceSulphaNgsGetDefaultConfig	24
sceSulphaNgsGetNeededMemory	25
sceSulphaNgsInit	20
sceSulphaNgsSetRackName	2 ⁻
sceSulphaNgsSetSampleName	28
sceSulphaNgsSetSynthName	29
sceSulphaNgsSetVoiceName	
sceSulphaNgsShutdown	3 ⁻
sceSulphaNgsTrace	32



Return Codes

Define	Description
SCE_SULPHA_ERROR_ALREADY_INITIALIZED	The system is already initialized.
SCE_SULPHA_ERROR_UNINITIALIZED	The system has not been initialized.
SCE_SULPHA_ERROR_INVALID_VALUE	A parameter is invalid.
SCE_SULPHA_ERROR_INVALID_POINTER	A pointer is invalid.
SCE_SULPHA_ERROR_INVALID_HANDLE	A handle is invalid.
SCE_SULPHA_ERROR_INVALID_ALIGNMENT	The alignment of a parameter is invalid.
SCE_SULPHA_ERROR_OUT_OF_MEMORY	The system ran out of memory.
SCE_SULPHA_ERROR_UNSUPPORTED	The feature is unsupported.
SCE_SULPHA_ERROR_CONNECTION	A network error occurred.
SCE_SULPHA_ERROR_NETWORK_OWNERSHIP	The network is not owned by this system.
SCE_SULPHA_ERROR_INVALID_AGENT_DEF	An agent has the incorrect definition.
SCE_SULPHA_ERROR_INVALID_AGENT_VERSION	An agent has the incorrect version number.
SCE_SULPHA_ERROR_INVALID_EVENT_ID	An invalid event ID was used.
SCE_SULPHA_ERROR_NOT_CONNECTED	The server is not connected.
SCE_SULPHA_ERROR_EVENT_ALREADY_SENT	An event has already been reported.
SCE_SULPHA_ERROR_AGENT_LIMIT_REACHED	The maximum number of agents has been reached.
SCE_SULPHA_ERROR_CAPTURE_STARTED	A capture is already in progress.
SCE_SULPHA_ERROR_OUT_OF_BOOKMARKS	The server has run out of bookmarks to allocate.
SCE_OK	Generic succeeded code.



SceSulphaConfig

Used to set up Sulpha behavior at initialization.

Definition

Members

notifyCallback A pointer to a user callback function of type SceSulphaCallback for receiving

client connection and disconnection notifications. Can be null.

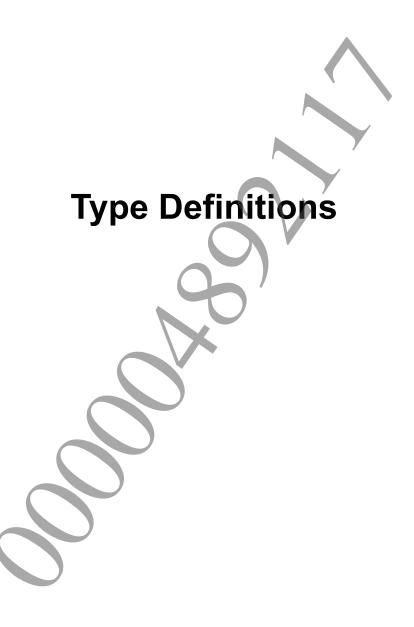
port Connection port.

bookmarkCount Maximum number of bookmarks available per update.

Description

Passed to Sulpha at initialization to configure the server.





SceSulphaUTF8

Data type for a Sulpha UTF8 string character.

Definition

#include <sulpha.h>
typedef SceUInt8 SceSulphaUTF8;

Description

Sulpha uses UTF8 strings, which allow for better localization support.



SceSulphaCallback

Sulpha connection callback.

Definition

Arguments

arg

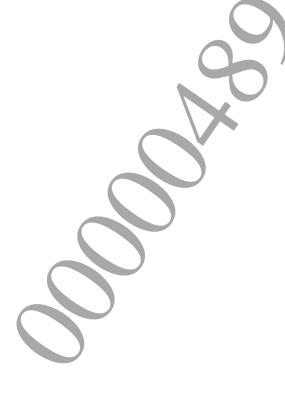
Currently unused.

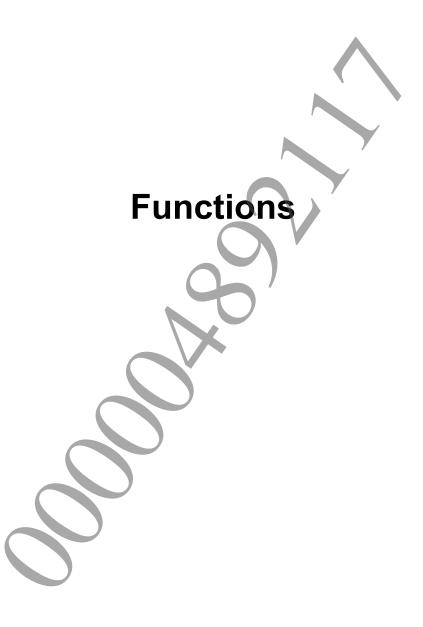
Return Values

None

Description

A typedef for the user callback function that can be registered in the $\underline{{\tt SceSulphaConfig}}$ structure. The function will be called when a client is connected or disconnected.





sceSulphaFileConnect

Starts capturing agent information directly to the specified file.

Definition

Arguments

filename

(Input) A fully qualified path name for the capture file.

Return Values

Value	Description
SCE OK	Successful completion.
SCE SULPHA ERROR UNINITIALIZED	The system has not been initialized.
SCE SULPHA ERROR INVALID POINTER	An invalid argument pointer was specified.
SCE SULPHA ERROR CAPTURE STARTED	A file capture is already in progress.

Calling Conditions

Can be called from an interrupt handler. Multithread safe.

Description

This function causes Sulpha to capture the agent information directly to the specified file instead of being sent over the network. The capture file can then be loaded into the Sulpha Tool at a later time. The user should still call sceSulphaUpdate() to periodically flush the captured data to the file.

See Also

sceSulphaFileDisconnect

sceSulphaFileDisconnect

Stops writing capture data to the file.

Definition

#include <sulpha.h>
int sceSulphaFileDisconnect(void);

Return Values

Value	Description
SCE OK	Successful completion,
SCE SULPHA ERROR UNINITIALIZED	The system has not been initialized.

Calling Conditions

Can be called from an interrupt handler.

Multithread safe.

Description

Stops writing capture data to the file and closes the file handle

See Also

sceSulphaFileConnect

sceSulphaGetDefaultConfig

Obtains the default Sulpha configuration structure.

Definition

Arguments

config

(Output) A pointer to a <u>SceSulphaConfig</u> configuration structure in which the default configuration will be stored.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE SULPHA ERROR INVALID POINTER	config is NULL.

Calling Conditions

Can be called from an interrupt handler.

Multithread safe.

Description

This function fills in the structure pointed to by <code>config</code> with the default configuration used to initialize Sulpha.

See Also

sceSulphaGetNeededMemory, sceSulphaInit

sceSulphaGetNeededMemory

Calculates total memory required to initialize the Sulpha system.

Definition

Arguments

config (Input) A <u>SceSulphaConfig</u> structure specifying system configuration

parameters.

sizeInBytes (Output) Pointer to a variable in which the required memory size is stored. This

value is in bytes.

Return Values

Value	Description
SCE OK	Successful completion.
SCE SULPHA ERROR INVALID POINTER	config or sizeInBytes is NULL.

Calling Conditions

Can be called from an interrupt handler.

Multithread safe.

Description

This function calculates the total memory required to initialize Sulpha using the <code>config</code> parameter. The calculated memory size is stored in the arguments pointer <code>sizeInBytes</code>. This value should be used to allocate memory to supply to the Sulpha system.

See Also

sceSulphaInit, sceSulphaGetDefaultConfig

sceSulphalnit

Initializes the Sulpha system.

Definition

Arguments

config
 buffer
 (Input) A <u>SceSulphaConfig</u> structure specifying required system resources.
 (Input) A pointer to a pre-allocated memory block of sufficient size to initialize the system. You can calculate the amount of memory required using the <u>sceSulphaGetNeededMemory()</u> function. Expressed in bytes.
 sizeInBytes
 (Input) The size of the pre-allocated memory block, expressed in bytes.

Return Values

Value	Description
SCE OK	Successful completion.
SCE SULPHA ERROR ALREADY INITIALIZED	The system is already initialized.
SCE SULPHA ERROR INVALID POINTER	config or buffer is NULL.
SCE SULPHA ERROR INVALID VALUE	sizeInBytes is invalid.
SCE SULPHA ERROR OUT OF MEMORY	The memory size required is less than available
	(supplied by the user).

Calling Conditions

Can be called from an interrupt handler. Multithread safe.

Description

This function initializes the Sulpha system. Before calling this function, you must determine the size of the system's memory requirement using the sceSulphaGetNeededMemory() function. Sulpha does not create any internal threads, therefore once Sulpha is initialized, the application should periodically call sceSulphaUpdate() to flush the captured data to file or network. The recommended frequency for flushing is the same as the synthesizer update.

See Also

sceSulphaGetNeededMemory, sceSulphaGetDefaultConfig, sceSulphaShutdown

sceSulphaNetworkInit

Network initialization helper.

Definition

#include <sulpha.h>
int sceSulphaNetworkInit(void);

Return Values

Value	Description
SCE OK	Successful completion.
SCE SULPHA ERROR CONNECTION	Network connection failure.
SCE SULPHA ERROR ALREADY INITIALIZED	The network is already initialized.

Calling Conditions

Can be called from an interrupt handler.

Multithread safe.

Description

This function should only be called under two circumstances: firstly, if the version of Sulpha for the target platform uses socket communications. Secondly, if there is no other network initialization processing by the user application.

See Also

sceSulphaNetworkShutdown



sceSulphaNetworkShutdown

Network shutdown helper.

Definition

#include <sulpha.h>
int sceSulphaNetworkShutdown(void);

Return Values

Value	Description
SCE OK	Successful completion.
SCE SULPHA ERROR UNINITIALIZED	The network was never initialized.

Calling Conditions

Can be called from an interrupt handler.

Multithread safe.

Description

This function should only be called if there is no other network initialization processing by the user application.

See Also

sceSulphaNetworkInit

sceSulphaSetBookmark

Sets a new bookmark with a given name and current timestamp and informs the Sulpha client.

Definition

Arguments

name (I

(Input) The name of the bookmark. Maximum length is 100 characters. (Input) The ID of the region that is to be created or -1 for no region.

Return Values

Value	Description
SCE OK	Successful completion.
SCE SULPHA ERROR OUT OF BOOKMARKS	The server has run out of bookmarks to allocate.
SCE SULPHA ERROR UNINITIALIZED	The system has not been initialized.
SCE SULPHA ERROR NOT CONNECTED	The server is not connected to a Sulpha client.

Calling Conditions

Can be called from an interrupt handler.
Multithread safe.

Description

This function adds a bookmark with the given name and current timestamp that can be viewed in the Sulpha Tool. It is also possible to supply bookmark identifiers that can be used as a bookmark regions if matching identifiers are found.

See Also

sceSulphaInit

sceSulphaShutdown

Shuts down the Sulpha system.

Definition

#include <sulpha.h>
int sceSulphaShutdown(void);

Return Values

Value	Description
SCE OK	Successful completion.
SCE SULPHA ERROR UNINITIALIZED	The system has not been initialized.

Calling Conditions

Can be called from an interrupt handler.

Multithread safe.

Description

This function shuts down the Sulpha system.

See Also

sceSulphaInit



Document serial number: 000004892117

sceSulphaUpdate

Updates the Sulpha system.

Definition

#include <sulpha.h>
int sceSulphaUpdate(void);

Return Values

Value	Description
SCE OK	Successful completion.
SCE SULPHA ERROR UNINITIALIZED	The system has not been initialized.

Calling Conditions

Can be called from an interrupt handler.

Multithread safe.

Description

This function updates the Sulpha system. It should be called at regular intervals to flush the captured data to file or network.

See Also

sceSulphaFileConnect





SceSulphaNgsConfig

Specifies the NGS agent configuration to calculate memory requirements.

Definition

Members

maxNamedObjects The maximum number of named objects that can be set.

maxTraceBufferBytes The maximum size of the message buffer in bytes.

Description

The application can obtain the default configuration by calling sceSulphaNgsGetDefaultConfig(). This structure should then be passed to the sceSulphaNgsGetNeededMemory() function to calculate the memory requirements. Finally this structure should be passed to sceSulphaNgsInit() to initialize the agent.





Document serial number: 000004892117

sceSulphaNgsGetDefaultConfig

Obtains the default NGS Agent configuration.

Definition

Arguments

config

(Output) Pointer to a structure in which the default configuration will be stored.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_SULPHA_ERROR_INVALID_POINTER	config is NULL.

Calling Conditions

Can be called from an interrupt handler. Multithread safe.

Description

This function fills in the structure pointed to by *config* with the default configuration used to initialize the NGS agent.

See Also

sceSulphaNgsInit, sceSulphaNgsGetNeededMemory

sceSulphaNgsGetNeededMemory

Calculates total memory required to initialize the Sulpha NGS Agent.

Definition

Arguments

config
sizeInBytes

(Input) A pointer to a structure describing the agent configuration.

(Output) Pointer to a size_t variable in which the needed memory size is stored. Value is in bytes.

Return Values

Value	Description
SCE OK	Successful completion.
SCE SULPHA ERROR INVALID POINTER	config or sizeInBytes is NULL.

Calling Conditions

Can be called from an interrupt handler. Multithread safe.

Description

This function calculates the total memory required to initialize the Sulpha NGS agent. The calculated memory size is stored in the <code>sizeInBytes</code> pointer. This value should be used to allocate memory to supply to the NGS agent.

See Also

sceSulphaNgsInit

sceSulphaNgsInit

Initializes the Sulpha NGS agent.

Definition

Arguments

config (Input) A pointer to a structure describing the agent configuration.

buffer (Input) A pointer to a pre-allocated memory block of sufficient size to initialize the

agent. You can calculate the amount of memory required using

the <u>sceSulphaNgsGetNeededMemory()</u> function.

sizeInBytes (Input) The size of the pre-allocated memory block, expressed in bytes.

Return Values

Value	Description
SCE OK	Successful completion.
SCE SULPHA ERROR INVALID POINTER	config or buffer is NULL.
SCE SULPHA ERROR INVALID VALUE	sizeInBytes is invalid.
SCE SULPHA ERROR ALREADY INITIALIZED	The system is already initialized.
SCE SULPHA ERROR UNINITIALIZED	The Sulpha system module has not been loaded.

Calling Conditions

Can be called from an interrupt handler. Multithread safe.

Description

Initializes the Sulpha NGS agent and registers itself with the Sulpha library. When initialized, the Sulpha NGS agent will be available to the user in the Sulpha Tool.

See Also

sceSulphaNgsGetNeededMemory, sceSulphaNgsShutdown

sceSulphaNgsSetRackName

Associates a user-defined name with the handle.

Definition

Arguments

rackHandle name (Input) A SceNgsHRack handle to be named.

(Input) A pointer to a null-terminated text string or NULL to clear.

The maximum name length is 27 characters. The name will be

truncated if it exceeds this size.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE SULPHA ERROR UNINITIALIZED	The agent has not been initialized.
SCE SULPHA ERROR INVALID HANDLE	rackHandle is invalid.

Description

Allows a user-defined name to be associated with the <code>rackHandle</code>. This name will be displayed in the Sulpha Tool. The name association can be removed by setting the <code>name</code> argument to NULL.

See Also

sceSulphaNgsSetSynthName, sceSulphaNgsSetVoiceName, sceSulphaNgsSetSampleName

sceSulphaNgsSetSampleName

Associates a user-defined name with the memory range.

Definition

Arguments

location length name (Input) The start address of the sample data to be named (Input) The size (in bytes) of the sample data in memory.

(Input) A pointer to a null-terminated text string or NULL to clear.

The maximum name length is 27 characters. The name will be

truncated if it exceeds this size.

Return Values

Value	Description
SCE OK	Successful completion.
SCE SULPHA ERROR UNINITIALIZED	The agent has not been initialized.
SCE SULPHA ERROR INVALID POINTER	location is NULL.
SCE SULPHA ERROR INVALID VALUE	length is 0.

Description

Allows a user-defined name to be associated with a range of memory. This name will be displayed in the Sulpha Tool when there is any activity involving this area. The name association can be removed by setting the <code>name</code> argument to NULL.

For example, if a gunshot PCM sample is stored at address 0×12345678 and its size is 0×4000 : sceSulphaNgsSetSampleName (0×12345678 , 0×4000 , "Gunshot");

Then, if any voice plays data from this area, the name "Gunshot" will be shown in the Sulpha Tool.

See Also

sceSulphaNgsSetSynthName, sceSulphaNgsSetRackName, sceSulphaNgsSetVoiceName

sceSulphaNgsSetSynthName

Associates a user-defined name with the handle.

Definition

```
#include <ngs/sulpha ngs.h>
int sceSulphaNgsSetSynthName(
        SceNgsHSynSystem synthHandle,
        const char *name
);
```

Arguments

synthHandle name

(Input) A SceNgsHSynSystem handle to be named.

(Input) A pointer to a null-terminated text string or NULL to clear. The maximum name length is 27 characters. The name will be truncated if it exceeds this size.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE SULPHA ERROR UNINITIALIZED	The agent has not been initialized.
SCE SULPHA ERROR INVALID HANDLE	synthHandle is invalid.

Description

Allows a user-defined name to be associated with the synthHandle. This name will be displayed in the Sulpha Tool. The name association can be removed by setting the name argument to NULL.

See Also

SulphaNgsSetVoiceName, sceSulphaNgsSetSampleName sceSulphaNgsSetRackName,

sceSulphaNgsSetVoiceName

Associates a user-defined name with the handle.

Definition

Arguments

voiceHandle
name

(Input) A SceNgsHVoice handle to be named.

(Input) A pointer to a null-terminated text string or NULL to clear.

The maximum name length is 27 characters. The name will be

truncated if it exceeds this size.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_SULPHA_ERROR_UNINITIALIZED	The agent has not been initialized.
SCE_SULPHA_ERROR_INVALID_HANDLE	voiceHandle is invalid.

Description

Allows a user-defined name to be associated with the *voiceHandle*. This name will be displayed in the Sulpha Tool. The name association can be removed by setting the *name* argument to NULL.

See Also

sceSulphaNgsSetSynthName, sceSulphaNgsSetRackName, sceSulphaNgsSetSampleName

sceSulphaNgsShutdown

Terminates the Sulpha NGS agent.

Definition

#include <ngs/sulpha_ngs.h>
int sceSulphaNgsShutdown();

Return Values

Value	Description
SCE OK	Successful completion.
SCE SULPHA ERROR UNINITIALIZED	The agent has not been initialized.

Description

This function shuts down the Sulpha NGS agent.

See Also

sceSulphaNgsInit



sceSulphaNgsTrace

Sends a text message to the Sulpha client.

Definition

Arguments

message

(Input) Pointer to a null-terminated text string. The maximum message length is 256 characters.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE SULPHA ERROR UNINITIALIZED	The agent has not been initialized.
SCE SULPHA ERROR INVALID POINTER	message is NULL.

Description

Adds a text string to the Sulpha data stream that allows meaningful messages to be displayed. For example, if the code is initializing a Voice resource, this function could send the text "Initializing GunShot voice".

