

# **libsulpha Reference**

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

## Table of Contents

<b>Return Codes .....</b>	<b>3</b>
Return Codes .....	4
<b>Structures .....</b>	<b>5</b>
SceSulphaConfig .....	6
<b>Type Definitions .....</b>	<b>7</b>
SceSulphaUTF8 .....	8
SceSulphaCallback .....	9
<b>Functions .....</b>	<b>10</b>
sceSulphaFileConnect .....	11
sceSulphaFileDisconnect .....	12
sceSulphaGetDefaultConfig .....	13
sceSulphaGetNeededMemory .....	14
sceSulphaInit .....	15
sceSulphaNetworkInit .....	16
sceSulphaNetworkShutdown .....	17
sceSulphaSetBookmark .....	18
sceSulphaShutdown .....	19
sceSulphaUpdate .....	20
<b>NGS Structures .....</b>	<b>21</b>
SceSulphaNgsConfig .....	22
<b>NGS Functions .....</b>	<b>23</b>
sceSulphaNgsGetDefaultConfig .....	24
sceSulphaNgsGetNeededMemory .....	25
sceSulphaNgsInit .....	26
sceSulphaNgsSetRackName .....	27
sceSulphaNgsSetSampleName .....	28
sceSulphaNgsSetSynthName .....	29
sceSulphaNgsSetVoiceName .....	30
sceSulphaNgsShutdown .....	31
sceSulphaNgsTrace .....	32

## Return Codes

000004892117

SCE CONFIDENTIAL

## 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.

# Structures

000004892117

SCE CONFIDENTIAL

---

# SceSulphaConfig

---

Used to set up Sulpha behavior at initialization.

## Definition

---

```
#include <sulpha.h>
struct SceSulphaConfig {
    SceSulphaCallback notifyCallback;
    SceUInt32 port;
    SceUInt32 bookmarkCount;
};
```

## Members

---

<i>notifyCallback</i>	A pointer to a user callback function of type <a href="#">SceSulphaCallback</a> for receiving client connection and disconnection notifications. Can be null.
<i>port</i>	Connection port.
<i>bookmarkCount</i>	Maximum number of bookmarks available per update.

## Description

---

Passed to Sulpha at initialization to configure the server.

# Type Definitions

000004892117

SCE CONFIDENTIAL

---

# 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.



SCE CONFIDENTIAL

---

# SceSulphaCallback

---

Sulpha connection callback.

## Definition

---

```
#include <sulpha.h>
typedef void (*SceSulphaCallback) (
    void *arg
);
```

## Arguments

---

<i>arg</i>	Currently unused.
------------	-------------------

## Return Values

---

None

## Description

---

A typedef for the user callback function that can be registered in the [SceSulphaConfig](#) structure. The function will be called when a client is connected or disconnected.

# Functions

000004892117

SCE CONFIDENTIAL

# sceSulphaFileConnect

Starts capturing agent information directly to the specified file.

## Definition

```
#include <sulpha.h>
int sceSulphaFileConnect (
    const char *filename
);
```

## Arguments

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

## Return Values

Value	Description
<a href="#">SCE_OK</a>	Successful completion.
<a href="#">SCE_SULPHA_ERROR_UNINITIALIZED</a>	The system has not been initialized.
<a href="#">SCE_SULPHA_ERROR_INVALID_POINTER</a>	An invalid argument pointer was specified.
<a href="#">SCE_SULPHA_ERROR_CAPTURE_STARTED</a>	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](#)

SCE CONFIDENTIAL

---

## sceSulphaFileDisconnect

---

Stops writing capture data to the file.

### Definition

---

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

### Return Values

---

Value	Description
<a href="#">SCE_OK</a>	Successful completion.
<a href="#">SCE_SULPHA_ERROR_UNINITIALIZED</a>	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](#)

SCE CONFIDENTIAL

## sceSulphaGetDefaultConfig

Obtains the default Sulpha configuration structure.

### Definition

```
#include <sulpha.h>
int sceSulphaGetDefaultConfig(
    SceSulphaConfig *config
);
```

### Arguments

*config* (Output) A pointer to a [SceSulphaConfig](#) configuration structure in which the default configuration will be stored.

### Return Values

Value	Description
<a href="#">SCE_OK</a>	Successful completion.
<a href="#">SCE_SULPHA_ERROR_INVALID_POINTER</a>	<i>config</i> 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 Sulpha.

### See Also

[sceSulphaGetNeededMemory](#), [sceSulphaInit](#)

SCE CONFIDENTIAL

# sceSulphaGetNeededMemory

Calculates total memory required to initialize the Sulpha system.

## Definition

```
#include <sulpha.h>
int sceSulphaGetNeededMemory (
    SceSulphaConfig const *config,
    SceUInt32 *sizeInBytes
);
```

## Arguments

*config* (Input) A [SceSulphaConfig](#) 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
<a href="#">SCE_OK</a>	Successful completion.
<a href="#">SCE_SULPHA_ERROR_INVALID_POINTER</a>	<i>config</i> or <i>sizeInBytes</i> 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 *config* parameter. The calculated memory size is stored in the arguments pointer *sizeInBytes*. This value should be used to allocate memory to supply to the Sulpha system.

## See Also

[sceSulphaInit](#), [sceSulphaGetDefaultConfig](#)

SCE CONFIDENTIAL

# sceSulphaInit

Initializes the Sulpha system.

## Definition

```
#include <sulpha.h>
int sceSulphaInit(
    SceSulphaConfig const *config,
    void *buffer,
    SceUInt32 sizeInBytes
);
```

## Arguments

<i>config</i>	(Input) A <a href="#">SceSulphaConfig</a> structure specifying required system resources.
<i>buffer</i>	(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 <a href="#">sceSulphaGetNeededMemory()</a> function. Expressed in bytes.
<i>sizeInBytes</i>	(Input) The size of the pre-allocated memory block, expressed in bytes.

## Return Values

Value	Description
<a href="#">SCE_OK</a>	Successful completion.
<a href="#">SCE_SULPHA_ERROR_ALREADY_INITIALIZED</a>	The system is already initialized.
<a href="#">SCE_SULPHA_ERROR_INVALID_POINTER</a>	<i>config</i> or <i>buffer</i> is NULL.
<a href="#">SCE_SULPHA_ERROR_INVALID_VALUE</a>	<i>sizeInBytes</i> is invalid.
<a href="#">SCE_SULPHA_ERROR_OUT_OF_MEMORY</a>	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](#)

SCE CONFIDENTIAL

---

# sceSulphaNetworkInit

---

Network initialization helper.

## Definition

---

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

## Return Values

---

Value	Description
<a href="#">SCE_OK</a>	Successful completion.
<a href="#">SCE_SULPHA_ERROR_CONNECTION</a>	Network connection failure.
<a href="#">SCE_SULPHA_ERROR_ALREADY_INITIALIZED</a>	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](#)



SCE CONFIDENTIAL

---

# sceSulphaNetworkShutdown

---

Network shutdown helper.

## Definition

---

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

## Return Values

---

Value	Description
<a href="#">SCE_OK</a>	Successful completion.
<a href="#">SCE_SULPHA_ERROR_UNINITIALIZED</a>	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](#)

SCE CONFIDENTIAL

# sceSulphaSetBookmark

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

## Definition

```
#include <sulpha.h>
int sceSulphaSetBookmark (
    SceSulphaUTF8 const *name,
    SceInt32 id
);
```

## Arguments

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

## Return Values

Value	Description
<a href="#">SCE_OK</a>	Successful completion.
<a href="#">SCE_SULPHA_ERROR_OUT_OF_BOOKMARKS</a>	The server has run out of bookmarks to allocate.
<a href="#">SCE_SULPHA_ERROR_UNINITIALIZED</a>	The system has not been initialized.
<a href="#">SCE_SULPHA_ERROR_NOT_CONNECTED</a>	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](#)

SCE CONFIDENTIAL

---

## sceSulphaShutdown

---

Shuts down the Sulpha system.

### Definition

---

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

### Return Values

---

Value	Description
<a href="#">SCE_OK</a>	Successful completion.
<a href="#">SCE_SULPHA_ERROR_UNINITIALIZED</a>	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](#)

SCE CONFIDENTIAL

---

# sceSulphaUpdate

---

Updates the Sulpha system.

## Definition

---

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

## Return Values

---

Value	Description
<a href="#">SCE_OK</a>	Successful completion.
<a href="#">SCE_SULPHA_ERROR_UNINITIALIZED</a>	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](#)

# NGS Structures

000004892117

---

# SceSulphaNgsConfig

---

Specifies the NGS agent configuration to calculate memory requirements.

## Definition

---

```
#include <ngs/sulpha_ngs.h>
struct SceSulphaNgsConfig {
    SceUInt32 maxNamedObjects;
    SceUInt32 maxTraceBufferBytes;
};
```

## Members

---

<i>maxNamedObjects</i>	The maximum number of named objects that can be set.
<i>maxTraceBufferBytes</i>	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.

# NGS Functions

000004892117

SCE CONFIDENTIAL

## sceSulphaNgsGetDefaultConfig

Obtains the default NGS Agent configuration.

### Definition

```
#include <ngs/sulpha_ngs.h>
int sceSulphaNgsGetDefaultConfig(
    SceSulphaNgsConfig *config
);
```

### 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	<i>config</i> 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

```
#include <ngs/sulpha_ngs.h>
int sceSulphaNgsGetNeededMemory(
    const SceSulphaNgsConfig *config,
    SceUInt32 *sizeInBytes
);
```

## Arguments

*config* (Input) A pointer to a structure describing the agent configuration.  
*sizeInBytes* (Output) Pointer to a `size_t` variable in which the needed memory size is stored.  
 Value is in bytes.

## Return Values

Value	Description
<a href="#">SCE_OK</a>	Successful completion.
<a href="#">SCE_SULPHA_ERROR_INVALID_POINTER</a>	<i>config</i> or <i>sizeInBytes</i> 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 *sizeInBytes* pointer. This value should be used to allocate memory to supply to the NGS agent.

## See Also

[sceSulphaNgsInit](#)

SCE CONFIDENTIAL

# sceSulphaNgsInit

Initializes the Sulpha NGS agent.

## Definition

```
#include <ngs/sulpha_ngs.h>
int sceSulphaNgsInit(
    const SceSulphaNgsConfig *config,
    void *buffer,
    SceUInt32 sizeInBytes
);
```

## Arguments

<i>config</i>	(Input) A pointer to a structure describing the agent configuration.
<i>buffer</i>	(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 <a href="#">sceSulphaNgsGetNeededMemory()</a> function.
<i>sizeInBytes</i>	(Input) The size of the pre-allocated memory block, expressed in bytes.

## Return Values

Value	Description
<a href="#">SCE_OK</a>	Successful completion.
<a href="#">SCE_SULPHA_ERROR_INVALID_POINTER</a>	<i>config</i> or <i>buffer</i> is NULL.
<a href="#">SCE_SULPHA_ERROR_INVALID_VALUE</a>	<i>sizeInBytes</i> is invalid.
<a href="#">SCE_SULPHA_ERROR_ALREADY_INITIALIZED</a>	The system is already initialized.
<a href="#">SCE_SULPHA_ERROR_UNINITIALIZED</a>	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](#)

SCE CONFIDENTIAL

# sceSulphaNgsSetRackName

Associates a user-defined name with the handle.

## Definition

```
#include <ngs/sulpha_ngs.h>
int sceSulphaNgsSetRackName(
    SceNgsHRack rackHandle,
    const char *name
);
```

## Arguments

*rackHandle* (Input) A *SceNgsHRack* handle to be named.  
*name* (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
<a href="#">SCE_OK</a>	Successful completion.
<a href="#">SCE_SULPHA_ERROR_UNINITIALIZED</a>	The agent has not been initialized.
<a href="#">SCE_SULPHA_ERROR_INVALID_HANDLE</a>	<i>rackHandle</i> is invalid.

## Description

Allows a user-defined name to be associated with the *rackHandle*. 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](#), [sceSulphaNgsSetVoiceName](#), [sceSulphaNgsSetSampleName](#)

SCE CONFIDENTIAL

# sceSulphaNgsSetSampleName

Associates a user-defined name with the memory range.

## Definition

```
#include <ngs/sulpha_ngs.h>
int sceSulphaNgsSetSampleName (
    const void *location,
    SceUInt32 length,
    const char *name
);
```

## Arguments

*location* (Input) The start address of the sample data to be named.  
*length* (Input) The size (in bytes) of the sample data in memory.  
*name* (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
<a href="#">SCE_OK</a>	Successful completion.
<a href="#">SCE_SULPHA_ERROR_UNINITIALIZED</a>	The agent has not been initialized.
<a href="#">SCE_SULPHA_ERROR_INVALID_POINTER</a>	<i>location</i> is NULL.
<a href="#">SCE_SULPHA_ERROR_INVALID_VALUE</a>	<i>length</i> 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 *name* argument to NULL.

For example, if a gunshot PCM sample is stored at address 0x12345678 and its size is 0x4000:  
 sceSulphaNgsSetSampleName(0x12345678, 0x4000, "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](#)

SCE CONFIDENTIAL

## sceSulphaNgsSetSynthName

Associates a user-defined name with the handle.

### Definition

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

### Arguments

*synthHandle* (Input) A *SceNgsHSynSystem* handle to be named.  
*name* (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
<a href="#">SCE_OK</a>	Successful completion.
<a href="#">SCE_SULPHA_ERROR_UNINITIALIZED</a>	The agent has not been initialized.
<a href="#">SCE_SULPHA_ERROR_INVALID_HANDLE</a>	<i>synthHandle</i> 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

[sceSulphaNgsSetRackName](#), [sceSulphaNgsSetVoiceName](#), [sceSulphaNgsSetSampleName](#)

SCE CONFIDENTIAL

# sceSulphaNgsSetVoiceName

Associates a user-defined name with the handle.

## Definition

```
#include <ngs/sulpha_ngs.h>
int sceSulphaNgsSetVoiceName(
    SceNgsHVoice voiceHandle,
    const char *name
);
```

## Arguments

*voiceHandle* (Input) A *SceNgsHVoice* handle to be named.  
*name* (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
<a href="#">SCE_OK</a>	Successful completion.
<a href="#">SCE_SULPHA_ERROR_UNINITIALIZED</a>	The agent has not been initialized.
<a href="#">SCE_SULPHA_ERROR_INVALID_HANDLE</a>	<i>voiceHandle</i> 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](#)

SCE CONFIDENTIAL

---

## sceSulphaNgsShutdown

---

Terminates the Sulpha NGS agent.

### Definition

---

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

### Return Values

---

Value	Description
<a href="#">SCE_OK</a>	Successful completion.
<a href="#">SCE_SULPHA_ERROR_UNINITIALIZED</a>	The agent has not been initialized.

### Description

---

This function shuts down the Sulpha NGS agent.

### See Also

---

[sceSulphaNgsInit](#)

SCE CONFIDENTIAL

---

## sceSulphaNgTrace

---

Sends a text message to the Sulpha client.

### Definition

---

```
#include <ngs/sulpha_ngs.h>
int sceSulphaNgTrace(
    const char *message
);
```

### Arguments

---

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

### Return Values

---

Value	Description
<a href="#">SCE_OK</a>	Successful completion.
<a href="#">SCE_SULPHA_ERROR_UNINITIALIZED</a>	The agent has not been initialized.
<a href="#">SCE_SULPHA_ERROR_INVALID_POINTER</a>	<i>message</i> 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".