NGSQuickSynth Library Reference

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Table of Contents

NGSQuickSynth Reference Overview

Introduction

This document describes the public members of the NGSQuickSynth Library API. For more information about this library and its components and dependencies, see the *NGSQuickSynth Library Overview* document, included in the SDK.

Error Definitions

This is the list of error codes that the NGSQuickSynth Library can generate:

Define	Value	Description
SCE_QUICK_SYNTH_OK	(0)	OK [0x00000000].
SCE_QUICK_SYNTH_ERROR_INVALID_PARAM	(-2137980927)	Invalid parameter [0x80910001].
SCE_QUICK_SYNTH_ERROR_ALLOC_FAIL	(-2137980926)	Memory allocation failed [0x80910002].
SCE_QUICK_SYNTH_ERROR_FREE_FAIL	(-2137980925)	Memory release failed [0x80910003].
SCE_QUICK_SYNTH_ERROR_NOT_INIT	(-2137980924)	Trying to access uninitialized memory [0x80910004].
SCE_QUICK_SYNTH_ERROR_PARSING_FILE	(-2137980923)	Expected element not found in XML configuration file [0x80910005].
SCE_QUICK_SYNTH_ERROR_INTERNAL	(-2137980922)	Internal error [0x80910006].
SCE_QUICK_SYNTH_ERROR_PARSING_CONFIG	(-2137980921)	Invalid configuration data in XML configuration file [0x80910007].
SCE_QUICK_SYNTH_ERROR_PARSING_SCENELIST	(-2137980920)	Invalid scene list configuration [0x80910008].
SCE_QUICK_SYNTH_ERROR_OUT_OF_ASSETS	(-2137980919)	Reached the maximum limit of allocated assets [0x80910009].
SCE_QUICK_SYNTH_ERROR_UNEXPECTED_XML	(-2137980918)	Unexpected item in XML file [0x8091000A].
SCE_QUICK_SYNTH_ERROR_PARSING_RACK	(-2137980917)	Incorrect rack configuration [0x8091000B].
SCE_QUICK_SYNTH_ERROR_PARSING_PATCH	(-2137980916)	Incorrect patch configuration [0x8091000C].
SCE_QUICK_SYNTH_ERROR_MISSING_ASSET	(-2137980915)	Configuration asset not found [0x8091000D].
SCE_QUICK_SYNTH_ERROR_PARSING_PRESETLIST	(-2137980914)	Invalid preset list configuration [0x8091000E].
SCE_QUICK_SYNTH_ERROR_PARSING_PRESET	(-2137980913)	Invalid preset configuration [0x8091000F].

The NGSQuickSynth library can return other error codes, if these are generated by other libraries in the SDK used by QuickSynth (for example, NGS). For a full list of error codes see the "Error Table" contained in the SDK in `\host_tools\debugging\error_code`.

Methods

QuickSynth Constructor

QuickSynth class constructor.

Definition

```
#include <quicksynth.h>
QuickSynth();
```

Calling Conditions

Multithread safe.

Description

QuickSynth class constructor.

QuickSynth Destructor

QuickSynth class destructor.

Definition

```
#include <quicksynth.h>
~QuickSynth();
```

Calling Conditions

Not multithread safe.

Description

The destructor method releases the resources allocated by QuickSynth.

initialise

QuickSynth initialization function.

Definition

```
#include <quicksynth.h>
SceInt32 initialise(
    const SceQuickSynthAllocatorFunc allocatorFunction,
    const SceQuickSynthFreeFunc freeFunction
);
```

Calling Conditions

Multithread safe.

Arguments

allocatorFunction	Memory allocator function e.g. `malloc()`
freeFunction	Memory deallocator function e.g. `free()`

Return Values

See Error Definitions.

Description

This method initializes the resources needed by the NGSQuickSynth library and must be called before using any other functionality in the class. It frees previously allocated assets, assigns pointers to memory management functions, allocates and initializes delegate objects.

If no memory management functions are passed as arguments, the default functions `malloc()` and `free()` are used.

release

QuickSynth releasing function.

Definition

```
#include <quicksynth.h>
SceInt32 release();
```

Calling Conditions

Multithread safe.

Return Values

See Error Definitions.

Description

This method releases the assets allocated by a QuickSynth object. It is called automatically when a QuickSynth object is destructed.

loadFromFile

Loads QuickSynth configuration from file.

Definition

```
#include <quicksynth.h>
SceInt32 loadFromFile(
    const char *pszFilename
);
```

Calling Conditions

Multithread safe.

Arguments

pszFilename	Configuration file name.

Return Values

See Error Definitions.

Description

This method loads a QuickSynth configuration file (for format details, refer to the *QuickSynth Tool User's Guide*).

The NGS system is initialized (if not previously initialized, or if any system parameters have changed), and the first (default) Scene is set.

loadFromMemory

Loads the QuickSynth configuration from memory.

Definition

```
#include <quicksynth.h>
SceInt32 loadFromMemory(
    const void *pMemory
);
```

Calling Conditions

Multithread safe.

Arguments

pMemory	Pointer to configuration data.

Return Values

See Error Definitions.

Description

This method loads the QuickSynth configuration (for data format details, refer to the *QuickSynth Tool User's Guide*). The NGS system is initialized (if not previously initialized, or if any system parameters have changed), and the first (default) Scene is set.

setScene (by scene name)

Reconfigures the NGS system as defined by the scene information.

Definition

```
#include <quicksynth.h>
SceInt32 setScene(
    const char *pszSceneName
);
```

Calling Conditions

Multithread safe.

Arguments

pszSceneName	Name of the scene.

Return Values

See Error Definitions.

Description

This method reconfigures NGS to match the scene, identified by the given name, including racks, presets, and patches configuration. The QuickSynth object reconfigures NGS as defined by Scenes in the QuickSynth configuration file.

setScene (by scene index)

Reconfigures the NGS system as defined by the scene information.

Definition

```
#include <quicksynth.h>
SceInt32 setScene(
    SceUInt32 uIndex
);
```

Calling Conditions

Multithread safe.

Arguments

uIndex	0-based index of the scene.

Return Values

See Error Definitions.

Description

This method reconfigures NGS to match the scene, identified by the given index, including racks, presets, and patches configuration. The QuickSynth object reconfigures NGS as defined by Scenes in the QuickSynth configuration file.

Note that there are two versions of `setScene`: by name and by index. To call this function by index, the argument must be of type `SceUInt32`. If 0 is passed without the correct type, it might be interpreted as a NULL string and therefore return an error.

getNumScenes

Gets the number of scenes available in the system.

Definition

```
#include <quicksynth.h>
SceInt32 getNumScenes() const;
```

Calling Conditions

Multithread safe.

Return Values

Returns the number of scenes or a negative value for errors. See Error Definitions.

Description

This function returns the number of scenes defined in the loaded QuickSynth configuration.

getNumRacksCurrentScene

Gets the number of racks in the scene currently initialized.

Definition

```
#include <quicksynth.h>
SceInt32 getNumRacksCurrentScene() const;
```

Calling Conditions

Multithread safe.

Return Values

Returns the number of racks or a negative value for errors. See Error Definitions.

Description

This function returns the number of racks defined in the scene currently loaded.

Document serial number: 000004892117

getNumPresets

Gets the number of presets defined in the current configuration.

Definition

```
#include <quicksynth.h>
SceInt32 getNumPresets(const char *pszRackType) const;
```

Calling Conditions

Can be called from an interrupt handler.

Multithread safe.

Arguments

pszRackType	Specify the name of a rack type to retrieve the number of presets defined for that rack type, or specify NULL to retrieve the total number of presets (for any rack types) in the current configuration.

Return Values

Returns the number of presets or a negative value for errors. See Error Definitions.

Description

This function returns the number of presets available in the system. By specifying the name of a rack type, the number of presets defined for the given rack type is returned. Alternatively, by specifying NULL for rack type, the total number of presets is returned.

getSystemInitParams

Gets the NGS System Initialization parameters.

Definition

```
#include <quicksynth.h>
const SceNgsSystemInitParams *getSystemInitParams() const;
```

Calling Conditions

Multithread safe.

Return Values

Returns a pointer to the `SceNgsSystemInitParams` structure used to initialize the NGS system or NULL if an error occurred.

Description

This function returns a pointer to the `SceNgsSystemInitParams` structure used to initialize the NGS system.

getSystemHandle

Gets the NGS System Handle.

Definition

```
#include <quicksynth.h>
SceNgsHSynSystem getSystemHandle() const;
```

Calling Conditions

Multithread safe.

Return Values

Returns the NGS system handle or `SCE_NGS_INVALID_HANDLE` if an error occurred.

Description

This function returns the NGS system handle, allowing the user to access NGS directly.

getRackHandle (by rack name)

Gets an NGS Rack Handle.

Definition

```
#include <quicksynth.h>
SceNgsHRack getRackHandle(const char *pszRackName) const;
```

Calling Conditions

Multithread safe.

Arguments

pszRackName	Rack name.

Return Values

Returns the NGS rack handle or `SCE_NGS_INVALID_HANDLE` if an error occurred.

Description

This function returns the NGS rack handle, given the name of the rack, allowing the user to access it directly.

getRackHandle (by rack index)

Gets an NGS Rack Handle.

Definition

```
#include <quicksynth.h>
SceNgsHRack getRackHandle(SceUInt32 uIndex) const;
```

Calling Conditions

Multithread safe.

Arguments

uIndex	0-based rack index.

Return Values

Returns the NGS rack handle or `SCE_NGS_INVALID_HANDLE` if an error occurred.

Description

This function returns the NGS rack handle, given the index of the rack, allowing the user to access it directly.

Note that there are two versions of `getRackHandle`: by name and by index. To call this function by index, the argument must be of type `SceUInt32`. If 0 is passed without the correct type, it might be interpreted as a NULL string and therefore return an error.

getRackDescription (by rack name)

Gets NGS Rack description structure.

Definition

```
#include <quicksynth.h>
const SceNgsRackDescription *getRackDescription(const char *pszRackName)
const;
```

Calling Conditions

Multithread safe.

Arguments

pszRackName	Rack name.

Return Values

Returns the NGS rack description structure or NULL if an error occurred.

Description

This function returns the `SceNgsRackDescription` structure used to initialize a NGS Rack, given the name of the rack. The function searches only the current Scene for a matching Rack.

getRackDescription (by rack index)

Gets NGS Rack description structure.

Definition

```
#include <quicksynth.h>
const SceNgsRackDescription *getRackDescription(SceUInt32 uIndex) const;
```

Calling Conditions

Multithread safe.

Arguments

uIndex	0-based rack index.

Return Values

Returns the NGS rack description structure or NULL if an error occurred.

Description

This function returns the `SceNgsRackDescription` structure used to initialize a NGS Rack, given the index of the rack. The function searches only the current Scene for a matching Rack.

Note that there are two versions of `getRackDescription`: by name and by index. To call this function by index, the argument must be of type `SceUInt32`. If 0 is passed without the correct type, it might be interpreted as a NULL string and therefore return an error.

getPreset (by preset name)

Gets an NGS Preset.

Definition

```
#include <quicksynth.h>
const SceNgsVoicePreset *getPreset(const char* pszPresetName) const;
```

Calling Conditions

Multithread safe.

Arguments

pszPresetName Preset name.

Return Values

Returns the `SceNgsVoicePreset` structure or NULL if an error occurred.

Description

This function, given the name of a preset, returns a structure containing the preset description (see `SceNgsVoicePreset` in the *NGS Reference* document), allowing the user to reconfigure the voices in the system.

getPreset (by preset index)

Gets an NGS Preset.

Definition

```
#include <quicksynth.h>
const SceNgsVoicePreset *getPreset(SceUInt32 uIndex) const;
```

Calling Conditions

Multithread safe.

Arguments

uIndex	0-based preset index.

Return Values

Returns the `SceNgsVoicePreset` structure or NULL if an error occurred.

Description

This function returns a structure containing the preset description (see `SceNgsVoicePreset` in the *NGS Reference* document), given the preset index, allowing the user to reconfigure the voices in the system.

Note that there are two versions of `getPreset`: by name and by index. To call this function by index, the argument must be of type `SceUInt32`. If 0 is passed without the correct type, it might be interpreted as a NULL string and therefore return an error.