

NP Trophy Library Reference

© 2015 Sony Computer Entertainment Inc.
All Rights Reserved.
SCE Confidential

Table of Contents

Initialization and Termination	4
sceNpTrophyInit	5
sceNpTrophyTerm	6
Handles	7
SceNpTrophyHandle	8
sceNpTrophyCreateHandle	9
sceNpTrophyDestroyHandle	10
sceNpTrophyAbortHandle	11
Contexts	12
SceNpTrophyContext	13
sceNpTrophyCreateContext	14
sceNpTrophyDestroyContext	16
Trophy Unlocking	17
sceNpTrophyUnlockTrophy	18
Unlock State Obtaining	20
SCE_NP_TROPHY_FLAG_ZERO	21
SCE_NP_TROPHY_FLAG_SET	22
SCE_NP_TROPHY_FLAG_SET_ALL	23
SCE_NP_TROPHY_FLAG_CLR	24
SCE_NP_TROPHY_FLAG_ISSET	25
SceNpTrophyFlagArray	26
sceNpTrophyGetTrophyUnlockState	27
Trophy Configuration Data/Trophy Record Obtaining	29
SceNpTrophyGameDetails	30
SceNpTrophyGameData	31
SceNpTrophyGroupId	32
SceNpTrophyGroupDetails	33
SceNpTrophyGroupData	34
SceNpTrophyGrade	35
SceNpTrophyDetails	36
SceNpTrophyData	37
sceNpTrophyGetGameInfo	38
sceNpTrophyGetGroupInfo	40
sceNpTrophyGetTrophyInfo	42
Icon Obtaining	44
sceNpTrophyGetGameIcon	45
sceNpTrophyGetGroupIcon	47
sceNpTrophyGetTrophyIcon	49
Trophy Setup Dialog	51
SceNpTrophySetupDialogParam	52
SceNpTrophySetupDialogResult	53
sceNpTrophySetupDialogParamInit	54
sceNpTrophySetupDialogInit	55

SCE CONFIDENTIAL

sceNpTrophySetupDialogTerm	57
sceNpTrophySetupDialogGetStatus	58
sceNpTrophySetupDialogGetResult	59
sceNpTrophySetupDialogAbort.....	60
Datatypes.....	61
SceNpTrophyId	62
Common Constants.....	63
Data Sizes	64
Trophy Setup Dialog Option.....	65
Return Codes	66

000004892117

Initialization and Termination

SCE CONFIDENTIAL

sceNpTrophyInit

Initialize the NP Trophy library

Definition

```
#include <np.h>
int sceNpTrophyInit(
    void *opt
)
```

Arguments

opt Options (reserved for future extension: always specify NULL)

Return Values

Returns SCE_OK (=0) for normal termination.

Returns a negative value for errors. The main error codes are shown below. (The application must not malfunction even if other error codes are returned.)

Value	(Number)	Description
SCE_NP_TROPHY_ERROR_ALREADY_INITIALIZED	0x80551602	Already initialized After calling <code>sceNpTrophyInit()</code> , <code>sceNpTrophyInit()</code> may have been called again without calling <code>sceNpTrophyTerm()</code> in between.
SCE_NP_TROPHY_ERROR_INVALID_ARGUMENT	0x80551604	Invalid argument

Description

This function initializes the NP Trophy library. Always call this function once before using the NP Trophy library.

Notes

This function is not multithread safe. When called from multiple threads at the same time, this function may terminate normally; however, subsequent operation of the library cannot be guaranteed. Please program your application so that this function is not called from multiple threads at the same time.

Examples

```
int ret;

ret = sceNpTrophyInit(NULL);
if (ret < 0) {
    // Error handling
}
```

See Also

`sceNpTrophyTerm()`

SCE CONFIDENTIAL

sceNpTrophyTerm

Terminate the NP Trophy library

Definition

```
#include <np.h>
int sceNpTrophyTerm(
    void
)
```

Arguments

None

Return Values

Returns SCE_OK (=0) for normal termination.

Returns a negative value for errors. The main error codes are shown below. (The application must not malfunction even if other error codes are returned.)

Value	(Number)	Description
SCE_NP_TROPHY_ERROR _NOT_INITIALIZED	0x80551601	Not initialized sceNpTrophyInit() has not been called.

Description

This function terminates the NP Trophy library.

Notes

Although the created context and handle will be automatically deleted when this function is called, it is recommended that the application explicitly delete them before calling this function.

This function is not multithread safe. When called from multiple threads at the same time, this function may terminate normally; however, subsequent operation of the library cannot be guaranteed. Please program your application so that this function is not called from multiple threads at the same time.

See Also

sceNpTrophyInit()

SCE CONFIDENTIAL

Handles

000004892117

Document serial number: 000004892117

SCE CONFIDENTIAL

SceNpTrophyHandle

Trophy handle

Definition

```
#include <np.h>
#define SCE_NP_TROPHY_INVALID_HANDLE (-1)
typedef SceInt32 SceNpTrophyHandle;
```

Description

This datatype represents a trophy handle.

A trophy handle is required to call an API and to abort its processing. Most of the NP Trophy library functions require a call to `sceNpTrophyCreateHandle()` first to create a handle, which must then be passed as an argument.

See Also

`sceNpTrophyCreateHandle()`, `sceNpTrophyDestroyHandle()`,
`sceNpTrophyAbortHandle()`

SCE CONFIDENTIAL

sceNpTrophyCreateHandle

Create a handle

Definition

```
#include <np.h>
int sceNpTrophyCreateHandle(
    SceNpTrophyHandle *handle
)
```

Arguments

handle Storage destination for the created handle

Return Values

Stores the created handle in **handle* and returns SCE_OK (=0) for normal termination.

Returns a negative value for errors. The main error codes are shown below. (The application must not malfunction even if other error codes are returned.)

Value	(Number)	Description
SCE_NP_TROPHY_ERROR_NOT_INITIALIZED	0x80551601	Not initialized <code>sceNpTrophyInit()</code> has not been called.
SCE_NP_TROPHY_ERROR_INVALID_ARGUMENT	0x80551604	Invalid argument
SCE_NP_TROPHY_ERROR_EXCEEDS_MAX	0x80551606	Exceeded the maximum number

Description

This function creates a handle to be used when calling an NP Trophy library function.

A handle can be reused. However, a handle must be destroyed if its processing has been aborted.

Examples

```
SceNpTrophyHandle handle = SCE_NP_TROPHY_INVALID_HANDLE;
int ret;

ret = sceNpTrophyCreateHandle(&handle);
if ( ret < 0 ) {
    // Error handling
}
```

See Also

`sceNpTrophyDestroyHandle()`

SCE CONFIDENTIAL

sceNpTrophyDestroyHandle

Destroy a handle

Definition

```
#include <np.h>
int sceNpTrophyDestroyHandle(
    SceNpTrophyHandle handle
)
```

Arguments

handle Handle to destroy

Return Values

Returns SCE_OK (=0) for normal termination.

Returns a negative value for errors. The main error codes are shown below. (The application must not malfunction even if other error codes are returned.)

Value	(Number)	Description
SCE_NP_TROPHY_ERROR_NOT_INITIALIZED	0x80551601	Not initialized <code>sceNpTrophyInit()</code> has not been called.
SCE_NP_TROPHY_ERROR_INVALID_HANDLE	0x80551608	Invalid handle

Description

This function destroys the handle specified by *handle* and frees the internal resources. A handle must be destroyed if its processing has been aborted.

See Also

`sceNpTrophyCreateHandle()`

SCE CONFIDENTIAL

sceNpTrophyAbortHandle

Abort a handle

Definition

```
#include <np.h>
int sceNpTrophyAbortHandle(
    SceNpTrophyHandle handle
)
```

Arguments

handle Handle to be aborted

Return Values

Returns SCE_OK (=0) for normal termination.

Returns a negative value for errors. The main error codes are shown below. (The application must not malfunction even if other error codes are returned.)

Value	(Number)	Description
SCE_NP_TROPHY_ERROR_NOT_INITIALIZED	0x80551601	Not initialized sceNpTrophyInit() has not been called.
SCE_NP_TROPHY_ERROR_INVALID_HANDLE	0x80551608	Invalid handle

Description

This function aborts the processing of an NP Trophy library function. For *handle*, specify the handle specified upon calling the function whose processing is to be aborted. The aborted function will return an error.

Destroy aborted handles without re-using them.

See Also

sceNpTrophyCreateHandle()

Contexts

000004892117

SCE CONFIDENTIAL

SceNpTrophyContext

Trophy context

Definition

```
#include <np.h>
#define SCE_NP_TROPHY_INVALID_CONTEXT (-1)
typedef SceInt32 SceNpTrophyContext;
```

Description

This datatype represents a trophy context. A trophy context contains information, such as, the NP Communication ID for identifying the trophy configuration data, and information representing the user for identifying his/her trophy record.

Most of the NP Trophy library functions require calling to `sceNpTrophyCreateContext()` in advance to create a context, then passing the context as an argument.

See Also

`sceNpTrophyCreateContext()`, `sceNpTrophyDestroyContext()`,
"Trophy Setup Dialog" chapter

SCE CONFIDENTIAL

sceNpTrophyCreateContext

Create a context

Definition

```
#include <np.h>
int sceNpTrophyCreateContext (
    SceNpTrophyContext *context,
    const SceNpCommunicationId *commId,
    const SceNpCommunicationSignature *commSign,
    SceUInt64 options
)
```

Arguments

context Storage destination for the created context
commId NP Communication ID
commSign NP communication signature
options Options (reserved for extension: always specify 0)

When NULL is specified to *commId* and *commSign*, the context is created using the NP Communication ID and the NP communication signature specified by `sceNpInit()`.

Return Values

Stores the created context in **context* and returns `SCE_OK (=0)` for normal termination.

Returns a negative value for errors. The main error codes are shown below. (The application must not malfunction even if other error codes are returned.)

Value	(Number)	Description
<code>SCE_NP_TROPHY_ERROR_NOT_INITIALIZED</code>	0x80551601	Not initialized <code>sceNpTrophyInit()</code> has not been called.
<code>SCE_NP_TROPHY_ERROR_INVALID_ARGUMENT</code>	0x80551604	Invalid argument
<code>SCE_NP_TROPHY_ERROR_EXCEEDS_MAX</code>	0x80551606	Exceeded the maximum number
<code>SCE_NP_TROPHY_ERROR_INVALID_NPCOMMID</code>	0x8055160a	Invalid NP Communication ID
<code>SCE_NP_TROPHY_ERROR_INVALID_NPCOMMSIGN</code>	0x8055160b	Invalid NP communication signature
<code>SCE_NP_TROPHY_ERROR_NPCOMMSIGN_VERIFICATION_FAILURE</code>	0x8055160c	Failed to verify the NP communication signature
<code>SCE_NP_TROPHY_ERROR_CONTEXT_ALREADY_EXISTS</code>	0x80551616	Context already exists Multiple contexts may have been created for an NP Communication ID.

Description

This function creates a context to be used when calling an NP Trophy library function.

The created context can be used until it is deleted by `sceNpTrophyDestroyContext()`. Even when a handle is aborted, there is no need to delete the context.

SCE CONFIDENTIAL

Notes

It is recommended that a context be used without deleting it as long as possible while the application is being used.

Normally, create a context upon application boot and set it up using Trophy Setup Dialog. Keep reusing it until the termination of the application.

Multiple contexts cannot be created for one NP Communication ID. Create one context per NP Communication ID.

Examples

```
extern SceNpCommunicationId commId;
extern SceNpCommunicationSignature commSign;

SceNpTrophyContext context = SCE_NP_TROPHY_INVALID_CONTEXT;
int ret;

ret = sceNpTrophyCreateContext(&context, &commId, &commSign, 0);
if ( ret < 0 ) {
    // Error handling
}
```

See Also

sceNpTrophyDestroyContext()

SCE CONFIDENTIAL

sceNpTrophyDestroyContext

Destroy a context

Definition

```
#include <np.h>
int sceNpTrophyDestroyContext(
    SceNpTrophyContext context
)
```

Arguments

context Context to destroy

Return Values

Returns SCE_OK (=0) for normal termination.

Returns a negative value for errors. The main error codes are shown below. (The application must not malfunction even if other error codes are returned.)

Value	(Number)	Description
SCE_NP_TROPHY_ERROR_NOT_INITIALIZED	0x80551601	Not initialized sceNpTrophyInit() has not been called.
SCE_NP_TROPHY_ERROR_INVALID_CONTEXT	0x80551609	Invalid context

Description

This function destroys the context specified with *context* and frees the internal resources.

Notes

It is recommended that a context be used without deleting it as long as possible while the application is being used.

See Also

sceNpTrophyCreateContext()

Trophy Unlocking

000004892117

SCE CONFIDENTIAL

sceNpTrophyUnlockTrophy

Unlock trophy

Definition

```
#include <np.h>
int sceNpTrophyUnlockTrophy(
    SceNpTrophyContext context,
    SceNpTrophyHandle handle,
    SceNpTrophyId trophyId,
    SceNpTrophyId *platinumId
)
```

Arguments

<i>context</i>	Context
<i>handle</i>	Handle
<i>trophyId</i>	Trophy ID to unlock
<i>platinumId</i>	Storage destination for the platinum trophy ID

Return Values

Stores the ID of the unlocked platinum trophy in **platinumId* if the platinum trophy is unlocked and returns SCE_OK (=0) for normal termination.

Returns a negative value for errors. The main error codes are shown below. (The application must not malfunction even if other error codes are returned.)

Value	(Number)	Description
SCE_NP_TROPHY_ERROR_NOT_INITIALIZED	0x80551601	Not initialized sceNpTrophyInit() has not been called.
SCE_NP_TROPHY_ERROR_INVALID_ARGUMENT	0x80551604	Invalid argument
SCE_NP_TROPHY_ERROR_INVALID_TROPHY_ID	0x8055160e	Invalid trophy ID
SCE_NP_TROPHY_ERROR_TROPHY_ALREADY_UNLOCKED	0x8055160f	The trophy has already been unlocked
SCE_NP_TROPHY_ERROR_PLATINUM_CANNOT_UNLOCK	0x80551610	Platinum trophy cannot be unlocked The platinum trophy is unlocked by the system and cannot be unlocked by this function.
SCE_NP_TROPHY_ERROR_SETUP_REQUIRED	0x80551612	Context has not been set up
SCE_NP_TROPHY_ERROR_UNLOCK_DENIED	0x8055161f	Save data does not contain the rights to obtain trophies

Description

This function unlocks the trophy specified by *trophyId* for the context specified by *context*. When this function is called, the specified trophy will be unlocked and it will be displayed in the trophy collection as an obtained trophy. When the trophy has been successfully unlocked, a notification message will be displayed.

If the requirements for unlocking the platinum trophy are met as a result of unlocking the specified trophy, the platinum trophy ID will be returned in **platinumId*. Otherwise, **platinumId* will return SCE_NP_TROPHY_INVALID_TROPHY_ID.

SCE CONFIDENTIAL

Notes

This function is blocking. Processing may take some time – make sure to call it from a subthread.

When the save data mounted at application start-up does not contain information on the current user's right to obtain trophies, the SCE_NP_TROPHY_ERROR_UNLOCK_DENIED error will occur and the trophy will not be unlocked.

Examples

```
extern SceNpTrophyContext context;
extern SceNpTrophyHandle handle;
extern SceNpTrophyId trophyId;
SceNpTrophyId platinumId = SCE_NP_TROPHY_INVALID_TROPHY_ID;
int ret;

ret = sceNpTrophyUnlockTrophy(
    context,
    handle,
    trophyId,
    &platinumId
);
if ( ret < 0 ) {
    // Error handling
}
if (platinumId != SCE_NP_TROPHY_INVALID_TROPHY_ID) {
    // Processing to be carried out when the platinum trophy is unlocked
}
```

See Also

```
sceNpTrophyCreateContext(), sceNpTrophyCreateHandle()
```

Unlock State Obtaining

000004892117

SCE CONFIDENTIAL

SCE_NP_TROPHY_FLAG_ZERO

Initialize trophy flag array

Definition

```
#include <np.h>
SCE_NP_TROPHY_FLAG_ZERO(p)
```

Arguments

p Pointer to SceNpTrophyFlagArray structure

Description

This macro initializes the trophy flag array.

It sets all the bits of the SceNpTrophyFlagArray structure specified with *p* to 0.

See Also

SceNpTrophyFlagArray, sceNpTrophyGetTrophyUnlockState()

SCE CONFIDENTIAL

SCE_NP_TROPHY_FLAG_SET

Set trophy flag

Definition

```
#include <np.h>
SCE_NP_TROPHY_FLAG_SET(n, p)
```

Arguments

n Trophy ID
p Pointer to SceNpTrophyFlagArray structure

Description

This macro adds certain trophy IDs to a trophy flag array.

It sets the bit corresponding to the trophy ID specified with *n* of the SceNpTrophyFlagArray structure specified with *p* to 1.

See Also

SceNpTrophyFlagArray, sceNpTrophyGetTrophyUnlockState()

SCE CONFIDENTIAL

SCE_NP_TROPHY_FLAG_SET_ALL

Fill trophy flag array

Definition

```
#include <np.h>
SCE_NP_TROPHY_FLAG_SET_ALL(p)
```

Arguments

p Pointer to SceNpTrophyFlagArray structure

Description

This macro fills the trophy flag array.

It sets all the bits of the SceNpTrophyFlagArray structure specified with *p* to 1.

See Also

SceNpTrophyFlagArray, sceNpTrophyGetTrophyUnlockState()

SCE CONFIDENTIAL

SCE_NP_TROPHY_FLAG_CLR

Clear trophy flags

Definition

```
#include <np.h>
SCE_NP_TROPHY_FLAG_CLR(n, p)
```

Arguments

n Trophy ID
p Pointer to SceNpTrophyFlagArray structure

Description

This macro removes certain trophy IDs from a trophy flag array.

It sets the bit corresponding to the trophy ID specified with *n* of the SceNpTrophyFlagArray structure specified with *p* to 0.

See Also

SceNpTrophyFlagArray, sceNpTrophyGetTrophyUnlockState()

SCE CONFIDENTIAL

SCE_NP_TROPHY_FLAG_ISSET

Examine trophy flag

Definition

```
#include <np.h>
SCE_NP_TROPHY_FLAG_ISSET(n, p)
```

Arguments

n Trophy ID
p Pointer to SceNpTrophyFlagArray structure

Return Values

Value	Description
Not 0	Trophy ID <i>n</i> is set (the bit is 1)
0	Trophy ID <i>n</i> is not set (the bit is 0)

Description

This macro checks whether or not a certain trophy flag is set.

It returns whether or not the bit corresponding to the trophy ID specified with *n* of the SceNpTrophyFlagArray structure specified with *p* is 1.

See Also

SceNpTrophyFlagArray, sceNpTrophyGetTrophyUnlockState()

SCE CONFIDENTIAL

SceNpTrophyFlagArray

Trophy flag array

Definition

```
#include <np.h>
#define SCE_NP_TROPHY_FLAG_SETSIZE      (128)
#define SCE_NP_TROPHY_FLAG_BITS_SHIFT   (5)
typedef uint32_t SceNpTrophyFlagMask;
typedef struct SceNpTrophyFlagArray {
    SceNpTrophyFlagMask flag_bits[
        SCE_NP_TROPHY_FLAG_SETSIZE >> SCE_NP_TROPHY_FLAG_BITS_SHIFT
    ];
} SceNpTrophyFlagArray;
```

Members

flag_bits Bits that represent an array of trophy IDs

Description

This structure maintains the trophy ID set to be examined by `sceNpTrophyGetTrophyUnlockState()`.

It contains bit flags specifying the trophy ID set, but it is not necessary to know the details of the structure, since normally the following macros are used for operation and evaluation.

Macro	Description
<code>SCE_NP_TROPHY_FLAG_ZERO(p)</code>	Initialize trophy flag array
<code>SCE_NP_TROPHY_FLAG_SET(n, p)</code>	Set trophy flag
<code>SCE_NP_TROPHY_FLAG_SET_ALL(p)</code>	Fill trophy flag array
<code>SCE_NP_TROPHY_FLAG_CLR(n, p)</code>	Clear trophy flags
<code>SCE_NP_TROPHY_FLAG_ISSET(n, p)</code>	Examine trophy flag

See Also

`SCE_NP_TROPHY_FLAG_ZERO()`, `SCE_NP_TROPHY_FLAG_SET()`,
`SCE_NP_TROPHY_FLAG_SET_ALL()`, `SCE_NP_TROPHY_FLAG_CLR()`,
`SCE_NP_TROPHY_FLAG_ISSET()`

SCE CONFIDENTIAL

sceNpTrophyGetTrophyUnlockState

Get trophy unlock state for all trophies

Definition

```
#include <np.h>
int sceNpTrophyGetTrophyUnlockState (
    SceNpTrophyContext context,
    SceNpTrophyHandle handle,
    SceNpTrophyFlagArray *flags,
    SceUInt32 *count
)
```

Arguments

<i>context</i>	Context
<i>handle</i>	Handle
<i>flags</i>	Storage destination for the obtained trophy record
<i>count</i>	Storage destination for the number of significant bits (total number of trophies) for the obtained trophy record

Return Values

Stores the trophy record in **flags*, stores the number of significant bits in **count*, and returns SCE_OK (=0) for normal termination.

Returns a negative value for errors. The main error codes are shown below. (The application must not malfunction even if other error codes are returned.)

Value	(Number)	Description
SCE_NP_TROPHY_ERROR_NOT_INITIALIZED	0x80551601	Not initialized sceNpTrophyInit () has not been called.
SCE_NP_TROPHY_ERROR_INVALID_ARGUMENT	0x80551604	Invalid argument
SCE_NP_TROPHY_ERROR_SETUP_REQUIRED	0x80551612	Context has not been set up

Description

This function obtains the record for all of the trophies in the context specified by *context*.

After calling this function, SCE_NP_TROPHY_FLAG_ISSET () can be used to determine from **flags* whether individual trophies were unlocked or not. In such cases, the maximum value that can be specified for the trophy ID will return to **count*.

Notes

This function is a blocking function. Because this function may take time to process, call it from a subthread.

SCE CONFIDENTIAL

Examples

```
extern SceNpTrophyContext context;
extern SceNpTrophyHandle handle;
SceNpTrophyFlagArray flags;
SceUInt32 count;
int ret;

SCE_NP_TROPHY_FLAG_ZERO(&flags);
count = 0;

ret = sceNpTrophyGetTrophyUnlockState(context, handle, &flags, &count);
if ( ret < 0 ) {
    // Error handling
}
```

See Also

sceNpTrophyCreateContext(), sceNpTrophyCreateHandle()

Trophy Configuration Data/Trophy Record Obtaining

SCE CONFIDENTIAL

SceNpTrophyGameDetails

Trophy configuration data of a trophy set

Definition

```
#include <np.h>
typedef struct SceNpTrophyGameDetails {
    SceSize size;
    SceUInt32 numGroups;
    SceUInt32 numTrophies;
    SceUInt32 numPlatinum;
    SceUInt32 numGold;
    SceUInt32 numSilver;
    SceUInt32 numBronze;
    SceChar8 title[SCE_NP_TROPHY_GAME_TITLE_MAX_SIZE];
    SceChar8 description[SCE_NP_TROPHY_GAME_DESCR_MAX_SIZE];
} SceNpTrophyGameDetails;
```

Members

<i>size</i>	Structure size (set sizeof(SceNpTrophyGameDetails))
<i>numGroups</i>	Defined total number of trophy groups
<i>numTrophies</i>	Defined total number of trophies
<i>numPlatinum</i>	Defined total number of platinum trophies
<i>numGold</i>	Defined total number of gold trophies
<i>numSilver</i>	Defined total number of silver trophies
<i>numBronze</i>	Defined total number of bronze trophies
<i>title</i>	Name of the trophy set
<i>description</i>	Description of the trophy set

Description

This structure is used to obtain trophy configuration data of a trophy set with `sceNpTrophyGetGameInfo()`.

The size of this structure must be set in *size* before passing it to `sceNpTrophyGetGameInfo()`.

This structure represents information that is fixed by the definition of the trophy set. Information that changes depending on user gameplay is represented by the `SceNpTrophyGameData` structure.

See Also

`sceNpTrophyGetGameInfo()`, `SceNpTrophyGameData`

SCE CONFIDENTIAL

SceNpTrophyGameData

Trophy record of a trophy set

Definition

```
#include <np.h>
typedef struct SceNpTrophyGameData {
    SceSize size;
    SceUInt32 unlockedTrophies;
    SceUInt32 unlockedPlatinum;
    SceUInt32 unlockedGold;
    SceUInt32 unlockedSilver;
    SceUInt32 unlockedBronze;
    SceUInt32 progressPercentage;
} SceNpTrophyGameData;
```

Members

<i>size</i>	Structure size (set sizeof (SceNpTrophyGameData))
<i>unlockedTrophies</i>	Number of unlocked trophies
<i>unlockedPlatinum</i>	Number of unlocked platinum trophies
<i>unlockedGold</i>	Number of unlocked gold trophies
<i>unlockedSilver</i>	Number of unlocked silver trophies
<i>unlockedBronze</i>	Number of unlocked bronze trophies
<i>progressPercentage</i>	Progress of the processing (%)

Description

This structure is used to obtain information on unlocked trophies (trophy record) of a trophy set with `sceNpTrophyGetGameInfo()`.

The size of this structure must be set in *size* before passing it to `sceNpTrophyGetGameInfo()`.

This structure represents information that changes depending on user gameplay. Fixed information defined by the trophy set is represented by the `SceNpTrophyGameDetails` structure.

See Also

`sceNpTrophyGetGameInfo()`, `SceNpTrophyGameDetails`

SCE CONFIDENTIAL

SceNpTrophyGroupId

Trophy group ID

Definition

```
#include <np.h>
#define SCE_NP_TROPHY_INVALID_GROUP_ID (-2)
#define SCE_NP_TROPHY_BASE_GAME_GROUP_ID (-1)
typedef SceInt32 SceNpTrophyGroupId;
```

Description

This datatype represents the trophy group ID.

A trophy group ID is a unique value specifying a single trophy group within a trophy set for a single title bound by an NP Communication ID.

SCE_NP_TROPHY_BASE_GAME_GROUP_ID corresponds to the base game section of the group ID.

See Also

`sceNpTrophyGetGroupInfo()`, `sceNpTrophyGetGroupIcon()`

SCE CONFIDENTIAL

SceNpTrophyGroupDetails

Trophy configuration data of a trophy group

Definition

```
#include <np.h>
typedef struct SceNpTrophyGroupDetails {
    SceSize size;
    SceNpTrophyGroupId groupId;
    SceUInt32 numTrophies;
    SceUInt32 numPlatinum;
    SceUInt32 numGold;
    SceUInt32 numSilver;
    SceUInt32 numBronze;
    SceChar8 title[SCE_NP_TROPHY_GROUP_TITLE_MAX_SIZE];
    SceChar8 description[SCE_NP_TROPHY_GROUP_DESCR_MAX_SIZE];
} SceNpTrophyGroupDetails;
```

Members

<i>size</i>	Structure size (set sizeof(SceNpTrophyGroupDetails))
<i>groupId</i>	Trophy group ID
<i>numTrophies</i>	Defined total number of trophies
<i>numPlatinum</i>	Defined total number of platinum trophies
<i>numGold</i>	Defined total number of gold trophies
<i>numSilver</i>	Defined total number of silver trophies
<i>numBronze</i>	Defined total number of bronze trophies
<i>title</i>	Name of the trophy group
<i>description</i>	Description of the trophy group

Description

This structure is used to obtain trophy configuration data of a trophy group with `sceNpTrophyGetGroupInfo()`.

The size of this structure must be set in *size* before passing it to `sceNpTrophyGetGroupInfo()`.

This structure represents information that is fixed by the definition of the trophy set. Information that changes depending on user gameplay is represented by the `SceNpTrophyGroupData` structure.

See Also

`sceNpTrophyGetGameInfo()`, `SceNpTrophyGroupData`

SCE CONFIDENTIAL

SceNpTrophyGroupData

Trophy record of a trophy group

Definition

```
#include <np.h>
typedef struct SceNpTrophyGroupData {
    SceSize size;
    SceNpTrophyGroupId groupId;
    SceUInt32 unlockedTrophies;
    SceUInt32 unlockedPlatinum;
    SceUInt32 unlockedGold;
    SceUInt32 unlockedSilver;
    SceUInt32 unlockedBronze;
    SceUInt32 progressPercentage;
} SceNpTrophyGroupData;
```

Members

<i>size</i>	Structure size (set sizeof (SceNpTrophyGroupData))
<i>groupId</i>	Trophy group ID
<i>unlockedTrophies</i>	Number of unlocked trophies
<i>unlockedPlatinum</i>	Number of unlocked platinum trophies
<i>unlockedGold</i>	Number of unlocked gold trophies
<i>unlockedSilver</i>	Number of unlocked silver trophies
<i>unlockedBronze</i>	Number of unlocked bronze trophies
<i>progressPercentage</i>	Progress of the processing (%)

Description

This structure is used to obtain information on unlocked trophies (trophy record) of a trophy group with `sceNpTrophyGetGroupInfo()`.

The size of this structure must be set in *size* before passing it to `sceNpTrophyGetGroupInfo()`.

This structure represents information that changes depending on user gameplay. Fixed information defined by the trophy set is represented by the `SceNpTrophyGroupDetails` structure.

See Also

`sceNpTrophyGetGroupInfo()`, `SceNpTrophyGroupDetails`

SCE CONFIDENTIAL

SceNpTrophyGrade

Trophy grade

Definition

```
#include <np.h>
typedef SceInt32 SceNpTrophyGrade;
```

Description

This datatype represents the grade of a trophy. The grade indicates the difficulty regarding the obtainment of a trophy.

Possible values are as follows.

Value	(Number)	Description
SCE_NP_TROPHY_GRADE_UNKNOWN	0	Grade unknown
SCE_NP_TROPHY_GRADE_PLATINUM	1	Platinum trophy: A trophy that is unlocked automatically by the system when all the required trophies are obtained
SCE_NP_TROPHY_GRADE_GOLD	2	Gold trophy: The most difficult trophy to obtain
SCE_NP_TROPHY_GRADE_SILVER	3	Silver trophy: A trophy that is moderately difficult to obtain
SCE_NP_TROPHY_GRADE_BRONZE	4	Bronze trophy: The most common type of trophy that can be easily obtained

See Also

SceNpTrophyDetails

SCE CONFIDENTIAL

SceNpTrophyDetails

Trophy configuration data of a trophy

Definition

```
#include <np.h>
typedef struct SceNpTrophyDetails {
    SceSize size;
    SceNpTrophyId trophyId;
    SceNpTrophyGrade trophyGrade;
    SceNpTrophyGroupId groupId;
    SceBool hidden;
    SceChar8 name[SCE_NP_TROPHY_NAME_MAX_SIZE];
    SceChar8 description[SCE_NP_TROPHY_DESCR_MAX_SIZE];
} SceNpTrophyDetails;
```

Members

<i>size</i>	Structure size (set sizeof(SceNpTrophyDetails))
<i>trophyId</i>	Trophy ID
<i>trophyGrade</i>	Grade of the trophy
<i>groupId</i>	Trophy group ID to which this trophy belongs
<i>hidden</i>	Hidden flag
<i>name</i>	Name of the trophy
<i>description</i>	Description of the trophy

Description

This structure is used to obtain trophy configuration data of a trophy with `sceNpTrophyGetTrophyInfo()`.

The size of this structure must be set in `size` before passing it to `sceNpTrophyGetTrophyInfo()`.

This structure represents information that is fixed by the definition of the trophy set. Information that changes depending on user gameplay is represented by the `SceNpTrophyData` structure.

See Also

`sceNpTrophyGetTrophyInfo()`, `SceNpTrophyData`

SCE CONFIDENTIAL

SceNpTrophyData

Trophy record of a trophy

Definition

```
#include <np.h>
typedef struct SceNpTrophyData {
    SceSize size;
    SceNpTrophyId trophyId;
    SceBool unlocked;
    SceUInt8 reserved[4];
    SceRtcTick timestamp;
} SceNpTrophyData;
```

Members

<i>size</i>	Structure size (set sizeof (SceNpTrophyData))
<i>trophyId</i>	Trophy ID
<i>unlocked</i>	Whether or not the trophy is unlocked
<i>reserved</i>	Reserved area (always fill with 0)
<i>timestamp</i>	The time stamp of when the trophy was first unlocked, or 0 if the trophy has not been unlocked

Description

This structure is used to obtain information on unlocked trophies (trophy record) of a trophy with `sceNpTrophyGetTrophyInfo()`.

The size of this structure must be set in *size* before passing it to `sceNpTrophyGetTrophyInfo()`.

timestamp represents the time stamp when the user obtained the trophy. Time stamps are recorded from when a user sets the time on the PlayStation®Vita via the Internet or signs in to PSN™.

The value of *timestamp* is set to 0 when the time stamp is not recorded, displaying ----/--/-- --:-- in the Trophies application of the system software.

This structure represents information that changes depending on user gameplay. Fixed information defined by the trophy set is represented by the `SceNpTrophyDetails` structure.

See Also

`sceNpTrophyGetTrophyInfo()`, `SceNpTrophyDetails`

SCE CONFIDENTIAL

sceNpTrophyGetGameInfo

Get trophy set information

Definition

```
#include <np.h>
int sceNpTrophyGetGameInfo (
    SceNpTrophyContext context,
    SceNpTrophyHandle handle,
    SceNpTrophyGameDetails *details,
    SceNpTrophyGameData *data
)
```

Arguments

<i>context</i>	Context
<i>handle</i>	Handle
<i>details</i>	Storage destination for the obtained trophy configuration data, or NULL
<i>data</i>	Storage destination for the obtained trophy records, or NULL

Return Values

Stores the obtained trophy configuration data in **details* (if *details* is not NULL), stores the obtained trophy records in **data* (if *data* is not NULL), and returns SCE_OK (=0) for normal termination.

Returns a negative value for errors. The main error codes are shown below. (The application must not malfunction even if other error codes are returned.)

Value	(Number)	Description
SCE_NP_TROPHY_ERROR_NOT_INITIALIZED	0x80551601	Not initialized <code>sceNpTrophyInit()</code> has not been called.
SCE_NP_TROPHY_ERROR_INVALID_ARGUMENT	0x80551604	Invalid argument
SCE_NP_TROPHY_ERROR_SETUP_REQUIRED	0x80551612	Context has not been set up

Description

This function obtains the trophy set information specified for *context*.

For **details*, static trophy configuration data determined by the definition of the trophy set will be returned. For **data*, user specific trophy records updated with the application calling an NP Trophy library function will be returned. If trophy configuration data or trophy records are not required, specify NULL for the corresponding argument. Specifying NULL in both *details* and *data* will return an error.

For *details->size* and *data->size*, set the size of each structure first and then call this function.

Notes

This function is a blocking function. Because this function may take time to process, call it from a subthread.

Examples

```
extern SceNpTrophyContext context;
extern SceNpTrophyHandle handle;
SceNpTrophyGameDetails details;
SceNpTrophyGameData data;
int ret;

memset(&details, 0x00, sizeof(details));
memset(&data, 0x00, sizeof(data));

details.size = sizeof(details);
data.size = sizeof(data);
ret = sceNpTrophyGetGameInfo(context, handle, &details, &data);
if ( ret < 0 ) {
    // Error handling
}
```

See Also

sceNpTrophyCreateContext(), sceNpTrophyCreateHandle()

SCE CONFIDENTIAL

sceNpTrophyGetGroupInfo

Get trophy group information

Definition

```
#include <np.h>
int sceNpTrophyGetGroupInfo (
    SceNpTrophyContext context,
    SceNpTrophyHandle handle,
    SceNpTrophyGroupId groupId,
    SceNpTrophyGroupDetails *details,
    SceNpTrophyGroupData *data
)
```

Arguments

<i>context</i>	Context
<i>handle</i>	Handle
<i>groupId</i>	Group ID
<i>details</i>	Storage destination for the obtained trophy configuration data, or NULL
<i>data</i>	Storage destination for the obtained trophy records, or NULL

Return Values

Stores the obtained trophy configuration data in **details* (if *details* is not NULL), stores the obtained trophy records in **data* (if *data* is not NULL), and returns SCE_OK (=0) for normal termination.

Returns a negative value for errors. The main error codes are shown below. (The application must not malfunction even if other error codes are returned.)

Value	(Number)	Description
SCE_NP_TROPHY_ERROR_NOT_INITIALIZED	0x80551601	Not initialized sceNpTrophyInit () has not been called.
SCE_NP_TROPHY_ERROR_INVALID_ARGUMENT	0x80551604	Invalid argument
SCE_NP_TROPHY_ERROR_INVALID_GROUP_ID	0x8055160d	Invalid group ID
SCE_NP_TROPHY_ERROR_SETUP_REQUIRED	0x80551612	Context has not been set up

Description

This function obtains the trophy group information specified with *context* and *groupId*.

Specify the ID of the trophy group you want to obtain information of to *groupId*. If you want to obtain information of a base game group, specify SCE_NP_TROPHY_BASE_GAME_GROUP_ID.

For **details*, static trophy configuration data determined by the definition of the trophy set will be returned. For **data*, user specific trophy records updated with the application calling an NP Trophy library function will be returned. If trophy configuration data or trophy records are not required, specify NULL for the corresponding argument. Specifying NULL in both *details* and *data* will return an error.

For *details->size* and *data->size*, set the size of each structure first and then call this function.

SCE CONFIDENTIAL

Notes

This function is a blocking function. Because this function may take time to process, call it from a subthread.

Examples

```
extern SceNpTrophyContext context;
extern SceNpTrophyHandle handle;
extern SceNpTrophyGroupId groupId;
SceNpTrophyGroupDetails details;
SceNpTrophyGroupData data;
int ret;

memset(&details, 0x00, sizeof(details));
memset(&data, 0x00, sizeof(data));

details.size = sizeof(details);
data.size = sizeof(data);
ret = sceNpTrophyGetGroupInfo(context, handle, groupId, &details, &data);
if ( ret < 0 ) {
    // Error handling
}
```

See Also

sceNpTrophyCreateContext(), sceNpTrophyCreateHandle()

SCE CONFIDENTIAL

sceNpTrophyGetTrophyInfo

Get trophy information

Definition

```
#include <np.h>
int sceNpTrophyGetTrophyInfo (
    SceNpTrophyContext context,
    SceNpTrophyHandle handle,
    SceNpTrophyId trophyId,
    SceNpTrophyDetails *details,
    SceNpTrophyData *data
)
```

Arguments

<i>context</i>	Context
<i>handle</i>	Handle
<i>trophyId</i>	Trophy ID
<i>details</i>	Storage destination for the obtained trophy configuration data, or NULL
<i>data</i>	Storage destination for the obtained trophy records, or NULL

Return Values

Stores the obtained trophy configuration data in **details* (if *details* is not NULL), stores the obtained trophy records in **data* (if *data* is not NULL), and returns SCE_OK (=0) for normal termination.

Returns a negative value for errors. The main error codes are shown below. (The application must not malfunction even if other error codes are returned.)

Value	(Number)	Description
SCE_NP_TROPHY_ERROR_NOT_INITIALIZED	0x80551601	Not initialized sceNpTrophyInit () has not been called.
SCE_NP_TROPHY_ERROR_INVALID_ARGUMENT	0x80551604	Invalid argument
SCE_NP_TROPHY_ERROR_INVALID_TROPHY_ID	0x8055160e	Invalid trophy ID
SCE_NP_TROPHY_ERROR_SETUP_REQUIRED	0x80551612	Context has not been set up

Description

This function obtains the trophy information specified with *context* and *groupId*.

For **details*, static trophy configuration data determined by the definition of the trophy set will be returned. For **data*, user specific trophy records updated with the application calling an NP Trophy library function will be returned. If trophy configuration data or trophy records are not required, specify NULL for the corresponding argument. Specifying NULL in both *details* and *data* will return an error.

For *details->size* and *data->size*, set the size of each structure first and then call this function.

Notes

This function is a blocking function. Because this function may take time to process, call it from a subthread.

©SCEI

SCE CONFIDENTIAL

Examples

```
extern SceNpTrophyContext context;
extern SceNpTrophyHandle handle;
extern SceNpTrophyId trophyId;
SceNpTrophyDetails details;
SceNpTrophyData data;
int ret;

memset(&details, 0x00, sizeof(details));
memset(&data, 0x00, sizeof(data));

details.size = sizeof(details);
data.size = sizeof(data);
ret = sceNpTrophyGetTrophyInfo(context, handle, trophyId, &details, &data);
if ( ret < 0 ) {
    // Error handling
}
```

See Also

sceNpTrophyCreateContext(), sceNpTrophyCreateHandle()

Icon Obtaining

000004892117

SCE CONFIDENTIAL

sceNpTrophyGetGameIcon

Get trophy set still-image icon

Definition

```
#include <np.h>
int sceNpTrophyGetGameIcon (
    SceNpTrophyContext context,
    SceNpTrophyHandle handle,
    void *buffer,
    SceSize *size
)
```

Arguments

<i>context</i>	Context
<i>handle</i>	Handle
<i>buffer</i>	Storage destination for the obtained icon data, or NULL
<i>size</i>	Size of <i>buffer</i> [] [IN]/Size of obtained icon data [OUT]

Return Values

Stores the obtained icon data in *buffer*[] (if *buffer* is not NULL), stores the icon data size in **size*, and returns SCE_OK (=0) for normal termination.

Returns a negative value for errors. The main error codes are shown below. (The application must not malfunction even if other error codes are returned.)

Value	(Number)	Description
SCE_NP_TROPHY_ERROR_NOT_INITIALIZED	0x80551601	Not initialized sceNpTrophyInit() has not been called.
SCE_NP_TROPHY_ERROR_INVALID_ARGUMENT	0x80551604	Invalid argument
SCE_NP_TROPHY_ERROR_SETUP_REQUIRED	0x80551612	Context has not been set up
SCE_NP_TROPHY_ERROR_ICON_FILE_NOT_FOUND	0x80551618	Icon file cannot be found

Description

This function obtains the trophy set still-image icon included in the trophy set for the specified *context*.

The icon data will be stored in the area specified by *buffer* as a PNG image. Prepare an appropriate buffer, then specify its address for *buffer* and specify its size for **size*.

In order to determine the exact buffer size necessary for storing the icon, call this function with NULL specified to *buffer* and a pointer to an appropriate variable specified to *size*. The necessary size will be calculated and stored to **size*.

Notes

This function is a blocking function. Because this function may take time to process, call it from a subthread.

Icon data installed in the internal flash memory is not referenced; instead, icon data in the trophy pack file, which is the install source, will be directly referenced.

Examples

```
extern SceNpTrophyContext context;
extern SceNpTrophyHandle handle;
int ret;
void *buf = NULL;
SceSize s = 0;

ret = sceNpTrophyGetGameIcon(context, handle, NULL, &s);
if ( ret < 0 ) {
    // Error handling
}
buf = malloc(s);
if (buf == NULL) {
    // Error handling
}
ret = sceNpTrophyGetGameIcon(context, handle, buf, &s);
if ( ret < 0 ) {
    // Error handling
}
```

See Also

sceNpTrophyCreateContext(), sceNpTrophyCreateHandle()

sceNpTrophyGetGroupIcon

Get trophy group still-image icon

Definition

```
#include <np.h>
int sceNpTrophyGetGroupIcon (
    SceNpTrophyContext context,
    SceNpTrophyHandle handle,
    SceNpTrophyGroupId groupId,
    void *buffer,
    SceSize *size
)
```

Argument

<i>context</i>	Context
<i>handle</i>	Handle
<i>groupId</i>	Group ID
<i>buffer</i>	Storage destination for the obtained icon data, or NULL
<i>size</i>	Size of <i>buffer</i> [] [IN]/Size of obtained icon data [OUT]

Return Values

Stores the obtained icon data in *buffer*[] (if *buffer* is not NULL), stores the icon data size in **size*, and returns SCE_OK (=0) for normal termination.

Returns a negative value for errors. The main error codes are shown below. (The application must not malfunction even if other error codes are returned.)

Value	(Number)	Description
SCE_NP_TROPHY_ERROR_NOT_INITIALIZED	0x80551601	Not initialized <i>sceNpTrophyInit()</i> has not been called.
SCE_NP_TROPHY_ERROR_INVALID_ARGUMENT	0x80551604	Invalid argument
SCE_NP_TROPHY_ERROR_INVALID_GROUP_ID	0x8055160d	Invalid group ID
SCE_NP_TROPHY_ERROR_SETUP_REQUIRED	0x80551612	Context has not been set up
SCE_NP_TROPHY_ERROR_ICON_FILE_NOT_FOUND	0x80551618	Icon file cannot be found

Description

This function obtains the still-image icon of the trophy group specified by *context* and *groupId*. Specify the ID of the trophy group you want to obtain information of to *groupId*. If you want to obtain information of the base game group, specify SCE_NP_TROPHY_BASE_GAME_GROUP_ID.

The icon data will be stored in the area specified by *buffer* as a PNG image. Prepare an appropriate buffer, then specify its address for *buffer* and specify its size for **size*.

In order to determine the exact buffer size necessary for storing the icon, call this function with NULL specified to *buffer* and a pointer to an appropriate variable specified to *size*. The necessary size will be calculated and stored to **size*.

SCE CONFIDENTIAL

Notes

This function is a blocking function. Because this function may take time to process, call it from a subthread.

Icon data installed in the internal flash memory is not referenced; instead, icon data in the trophy pack file, which is the install source, will be directly referenced.

Examples

```
extern SceNpTrophyContext context;
extern SceNpTrophyHandle handle;
extern SceNpTrophyGroupId groupId;
int ret;
void *buf = NULL;
SceSize s = 0;

ret = sceNpTrophyGetGroupIcon(context, handle, groupId, NULL, &s);
if ( ret < 0 ) {
    // Error handling
}
buf = malloc(s);
if (buf == NULL) {
    // Error handling
}
ret = sceNpTrophyGetGroupIcon(context, handle, groupId, buf, &s);
if ( ret < 0 ) {
    // Error handling
}
```

See Also

sceNpTrophyCreateContext(), sceNpTrophyCreateHandle()

SCE CONFIDENTIAL

sceNpTrophyGetTrophyIcon

Get trophy still-image icon

Definition

```
#include <np.h>
int sceNpTrophyGetTrophyIcon (
    SceNpTrophyContext context,
    SceNpTrophyHandle handle,
    SceNpTrophyId trophyId,
    void *buffer,
    SceSize *size
)
```

Arguments

<i>context</i>	Context
<i>handle</i>	Handle
<i>trophyId</i>	Trophy ID
<i>buffer</i>	Storage destination for the obtained icon data, or NULL
<i>size</i>	Size of <i>buffer</i> [IN]/Size of obtained icon data [OUT]

Return Values

Stores the obtained icon data in *buffer* (if *buffer* is not NULL), stores the icon data size in **size*, and returns SCE_OK (=0) for normal termination.

Returns a negative value for errors. The main error codes are shown below. (The application must not malfunction even if other error codes are returned.)

Value	(Number)	Description
SCE_NP_TROPHY_ERROR_NOT_INITIALIZED	0x80551601	Not initialized <i>sceNpTrophyInit()</i> has not been called.
SCE_NP_TROPHY_ERROR_INVALID_ARGUMENT	0x80551604	Invalid argument
SCE_NP_TROPHY_ERROR_INVALID_TROPHY_ID	0x8055160e	Invalid trophy ID
SCE_NP_TROPHY_ERROR_SETUP_REQUIRED	0x80551612	Context has not been set up
SCE_NP_TROPHY_ERROR_ICON_FILE_NOT_FOUND	0x80551618	Icon file cannot be found
SCE_NP_TROPHY_ERROR_TROPHY_NOT_UNLOCKED	0x8055161e	The trophy has not been unlocked yet

Description

This function obtains the still-image icon of the trophy specified by *context* and *trophyId*.

The icon data is stored in the area specified by *buffer* as a PNG image. Prepare an appropriate buffer, then specify its address for *buffer* and specify its size for **size*.

In order to determine the exact buffer size necessary for storing the icon, call this function with NULL specified to *buffer* and a pointer to an appropriate variable specified to *size*. The necessary size will be calculated and stored to **size*.

SCE CONFIDENTIAL

Notes

This function is a blocking function. Because this function may take time to process, call it from a subthread.

Icon data installed in a memory card is not referenced; instead, icon data in the trophy pack file, which is the install source, will be directly referenced. Thus, icons can be obtained even when a memory card is not inserted.

A still-image icon cannot be obtained for a trophy that has not been earned by the user yet. Prepare an icon indicating a locked trophy, for example, and use it. Design icons in alignment to the application screen. It is also possible to display separate icons for hidden trophies and those that are not hidden.

Examples

```
extern SceNpTrophyContext context;
extern SceNpTrophyHandle handle;
extern SceNpTrophyId trophyId;
int ret;
void *buf = NULL;
SceSize s = 0;

ret = sceNpTrophyGetTrophyIcon(context, handle, trophyId, NULL, &s);
if ( ret < 0 ) {
    // Error handling
}
buf = malloc(s);
if (buf == NULL) {
    // Error handling
}
ret = sceNpTrophyGetTrophyIcon(context, handle, trophyId, buf, &s);
if ( ret < 0 ) {
    // Error handling
}
```

See Also

`sceNpTrophyCreateContext()`, `sceNpTrophyCreateHandle()`

Trophy Setup Dialog

000004892117

SCE CONFIDENTIAL

SceNpTrophySetupDialogParam

Boot parameters of Trophy Setup Dialog

Definition

```
#include <np_trophy_setup_dialog.h>
typedef struct SceNpTrophySetupDialogParam {
    SceUInt32 sdkVersion;
    SceCommonDialogParam commonParam;
    SceNpTrophyContext context;
    SceUInt32 options;
    SceUInt8 reserved[128];
} SceNpTrophySetupDialogParam;
```

Members

<i>sdkVersion</i>	Area used within the library. The application need not directly reference this space.
<i>commonParam</i>	Common parameters for Common Dialogs
<i>context</i>	Context to set up trophy
<i>options</i>	Options
<i>reserved</i>	Unused

Description

This datatype represents the boot parameters of Trophy Setup Dialog.

It is specified upon initializing the boot parameters with `sceNpTrophySetupDialogParamInit()` and starting Trophy Setup Dialog with `sceNpTrophySetupDialogInit()`.

Trophy Setup Dialog option can be specified for *options*. For the option that can be specified, refer to the "Trophy Setup Dialog Option" section. When it is not necessary to specify the option, do not specify anything because *options* is initialized to 0 with `sceNpTrophySetupDialogParamInit()`.

Trophy Setup Dialog does not support the background color specification; therefore, specify NULL to the *bgColor* argument of common parameters for Common Dialogs. For other common parameters for Common Dialogs, refer to the "Common Dialog Reference" document.

See Also

`sceNpTrophySetupDialogParamInit()`

SCE CONFIDENTIAL

SceNpTrophySetupDialogResult

Structure to store the result of Trophy Setup Dialog

Definition

```
#include <np_trophy_setup_dialog.h>
typedef struct SceNpTrophySetupDialogResult {
    SceInt32 result;
    SceUInt8 reserved[128];
} SceNpTrophySetupDialogResult;
```

Members

result Completion status of Trophy Setup Dialog
reserved Unused

Description

This datatype represents the processing result of Trophy Setup Dialog.
It is specified in `sceNpTrophySetupDialogGetResult()`.

See Also

`sceNpTrophySetupDialogGetResult()`

SCE CONFIDENTIAL

sceNpTrophySetupDialogParamInit

Initialize boot parameters of Trophy Setup Dialog

Definition

```
#include <np_trophy_setup_dialog.h>
static inline
void sceNpTrophySetupDialogParamInit (SceNpTrophySetupDialogParam *param)
{
    sceClibMemset( param, 0x0, sizeof(SceNpTrophySetupDialogParam) );
    _sceCommonDialogSetMagicNumber( &param->commonParam );
    param->sdkVersion = SCE_PSP2_SDK_VERSION;
    param->context = SCE_NP_TROPHY_INVALID_CONTEXT;
    param->options = 0;
}
```

Arguments

param Boot parameters

Return Values

None

Description

This function initializes the boot parameters specified with `sceNpTrophySetupDialogInit()`. This function must be called to initialize the boot parameters before `sceNpTrophySetupDialogInit()` is called.

The *context* member of the boot parameters will not be set by this function. Explicitly set it after calling this function.

See Also

`sceNpTrophySetupDialogInit()`

sceNpTrophySetupDialogInit

Start Trophy Setup Dialog

Definition

```
#include <np_trophy_setup_dialog.h>
SceInt32 sceNpTrophySetupDialogInit(
    SceNpTrophySetupDialogParam *param
);
```

Arguments

param Boot parameters

Return Values

Returns a value of 0 or above indicating the processing status common to the Common Dialog for normal termination.

Returns a negative value for an error.

Description

This function starts the processing of Trophy Setup Dialog. This function is a non-blocking function; it requests the start of Trophy Setup Dialog to the system and returns immediately.

After calling this function, call `sceNpTrophySetupDialogGetStatus()` to obtain the operation status. While setup is being processed, the operation status will be `SCE_COMMON_DIALOG_STATUS_RUNNING`. After this, when setup succeeds or when an error occurs, the operation status will be updated to `SCE_COMMON_DIALOG_STATUS_FINISHED`. The return value of the setup can be obtained with `sceNpTrophySetupDialogGetResult()`.

Regardless of the success/failure of the setup, call `sceNpTrophySetupDialogTerm()` to terminate the setup process once the operation status becomes `SCE_COMMON_DIALOG_STATUS_FINISHED`.

By default, Trophy Setup Dialog does not display anything while setup process is running. Only when an error occurs during the process or if a confirmation to the user is required, a dialog will be displayed. By specifying `SCE_NP_TROPHY_SETUP_DIALOG_OPTION_SHOW_PROCESSING_DIALOG` to the *options* member of the boot parameters of Trophy Setup Dialog, a dialog can be displayed during the setup process.

Notes

There is a limit to the number of trophy sets that can be saved on the internal flash memory. When this limit is exceeded, the oldest trophy set will be deleted first. In this case, a confirmation dialog will be displayed to the user to gain permission of the deletion.

Errors that occur when Trophy Setup Dialog calls fail will be errors that occur only during development such as argument specification mistakes. Therefore, the minimum error handling can be used in the application.

On the other hand, the application does not have to handle errors which occur after the call of Trophy Setup Dialog succeeds. An error dialog will be automatically displayed and the game cannot be continued.

By setting **Trophy Setup Dialog Debug** of ★**Debug Settings** to **Always Fail**, the operations performed when an error occurs can be emulated and checked.

SCE CONFIDENTIAL

See Also

SCE_COMMON_DIALOG_STATUS_XXX

sceNpTrophySetupDialogGetStatus(), sceNpTrophySetupDialogGetResult(),
sceNpTrophySetupDialogTerm()

000004892117

SCE CONFIDENTIAL

sceNpTrophySetupDialogTerm

Terminate Trophy Setup Dialog

Definition

```
#include <np_trophy_setup_dialog.h>
SceInt32 sceNpTrophySetupDialogTerm(
    void
);
```

Arguments

None

Return Values

Returns a value of 0 or above indicating the processing status common to the Common Dialog for normal termination.

Returns a negative value for an error.

Description

This function terminates Trophy Setup Dialog.

Make sure to call this function when setup processing of trophy information completes and SCE_COMMON_DIALOG_STATUS_FINISHED returns.

See Also

sceNpTrophySetupDialogInit()

sceNpTrophySetupDialogGetStatus

Get the operation status of Trophy Setup Dialog

Definition

```
#include <np_trophy_setup_dialog.h>
SceCommonDialogStatus sceNpTrophySetupDialogGetStatus (
    void
);
```

Arguments

None

Return Values

Returns a value of 0 or above indicating the processing status common to the Common Dialog for normal termination.

Returns a negative value for an error.

Description

This function returns the current operating status of Trophy Setup Dialog.

Value	(Number)	Description
SCE_COMMON_DIALOG_STATUS_NONE	0	Not processed
SCE_COMMON_DIALOG_STATUS_RUNNING	1	Processing
SCE_COMMON_DIALOG_STATUS_FINISHED	2	Processing complete

After calling `sceNpTrophySetupDialogInit()`, leave a certain interval and then call this function to obtain the operation status.

See Also

```
SCE_COMMON_DIALOG_STATUS_XXX
sceNpTrophySetupDialogInit()
```

SCE CONFIDENTIAL

sceNpTrophySetupDialogGetResult

Get result of Trophy Setup Dialog

Definition

```
#include <np_trophy_setup_dialog.h>
SceInt32 sceNpTrophySetupDialogGetResult(
    SceNpTrophySetupDialogResult *result
);
```

Arguments

result Pointer to the structure for storing the result

Return Values

Returns a value of 0 or above indicating the processing status common to the Common Dialog for normal termination.

Returns a negative value for an error.

Description

After the processing of Trophy Setup Dialog completes, call this function to obtain the result.

Value	(Number)	Description
SCE_COMMON_DIALOG_RESULT_OK	0	Normal termination
SCE_COMMON_DIALOG_RESULT_USER_CANCELED	1	User canceled the processing
SCE_COMMON_DIALOG_RESULT_ABORTED	2	Forced-termination by <code>sceNpTrophySetupDialogAbort()</code>
Negative value		Error termination Value represents the error code.

The result will be returned to *result* specified in the argument.

See Also

SCE_COMMON_DIALOG_RESULT_XXX
SceNpTrophySetupDialogResult

sceNpTrophySetupDialogAbort

Abort Trophy Setup Dialog

Definition

```
#include <np_trophy_setup_dialog.h>
SceInt32 sceNpTrophySetupDialogAbort(
    void
);
```

Arguments

None

Return Values

Returns a value of 0 or above indicating the processing status common to the Common Dialog for normal termination.

Returns a negative value for an error.

Description

This function aborts Trophy Setup Dialog.

This function can be called at any time after `sceNpTrophySetupDialogInit()` and before `sceNpTrophySetupDialogTerm()`.

When the call succeeds, Trophy Setup Dialog will begin to abort its processing. When the abort completes, the operation status will transition to `SCE_COMMON_DIALOG_STATUS_FINISHED`.

For details on the operation status, refer to the explanation of `sceNpTrophySetupDialogGetStatus()`.

When this function is called and Trophy Setup Dialog is aborted without being set up, the application will be terminated.

Datatypes

000004892117

SCE CONFIDENTIAL

SceNpTrophyId

Trophy ID

Definition

```
#include <np.h>
#define SCE_NP_TROPHY_INVALID_TROPHY_ID (-1)
typedef SceInt32 SceNpTrophyId;
```

Description

This datatype represents a trophy ID.

A trophy ID is a unique value that identifies a single trophy in a trophy set for a single title identified with an NP Communication ID.

See Also

```
sceNpTrophyGetTrophyInfo(), sceNpTrophyGetTrophyIcon(),
sceNpTrophyUnlockTrophy()
```

Common Constants

000004892117

Data Sizes

Sizes of various data used in the NP Trophy library

Definition

Value	(Number)	Description
SCE_NP_TROPHY_GAME_TITLE_MAX_SIZE	128	The maximum size for the name of a trophy set (including the NULL terminator)
SCE_NP_TROPHY_GAME_DESCR_MAX_SIZE	1024	The maximum size for the description of a trophy set (including the NULL terminator)
SCE_NP_TROPHY_GROUP_TITLE_MAX_SIZE	128	The maximum size for the name of a trophy group (including the NULL terminator)
SCE_NP_TROPHY_GROUP_DESCR_MAX_SIZE	1024	The maximum size for the description of a trophy group (including the NULL terminator)
SCE_NP_TROPHY_NAME_MAX_SIZE	128	The maximum size for the name of a trophy (including the NULL terminator)
SCE_NP_TROPHY_DESCR_MAX_SIZE	1024	The maximum size for the description of a trophy (including the NULL terminator)
SCE_NP_TROPHY_NUM_MAX	128	The maximum number of trophies that can be defined

Description

These are size definitions of data used in the NP Trophy library.

SCE CONFIDENTIAL

Trophy Setup Dialog Option

Trophy Setup Dialog option

Definition

Value	(Number)	Description
SCE_NP_TROPHY_SETUP_DIALOG_OPTION_ SHOW_PROCESSING_DIALOG	0x00000001	Displays a dialog while trophy setup process is running

Description

This is a definition of the option value used in Trophy Setup Dialog.

SCE CONFIDENTIAL

Return Codes

List of return codes returned by the NP Trophy library

Definition

Value	(Number)	Description
SCE_NP_TROPHY_ERROR_UNKNOWN	0x80551600	Undefined error, not listed below
SCE_NP_TROPHY_ERROR_NOT_INITIALIZED	0x80551601	Not initialized sceNpTrophyInit() has not been called.
SCE_NP_TROPHY_ERROR_ALREADY_INITIALIZED	0x80551602	Already initialized After calling sceNpTrophyInit(), sceNpTrophyInit() may have been called again without sceNpTrophyTerm() being called in between.
SCE_NP_TROPHY_ERROR_NO_MEMORY	0x80551603	Not enough memory
SCE_NP_TROPHY_ERROR_INVALID_ARGUMENT	0x80551604	Invalid argument
SCE_NP_TROPHY_ERROR_INSUFFICIENT_BUFFER	0x80551605	Insufficient buffer to continue processing
SCE_NP_TROPHY_ERROR_EXCEEDS_MAX	0x80551606	Exceeded the maximum value
SCE_NP_TROPHY_ERROR_ABORT	0x80551607	Handle has been aborted Processing has been aborted by sceNpTrophyAbortHandle(). Delete the handle.
SCE_NP_TROPHY_ERROR_INVALID_HANDLE	0x80551608	Invalid handle
SCE_NP_TROPHY_ERROR_INVALID_CONTEXT	0x80551609	Invalid context
SCE_NP_TROPHY_ERROR_INVALID_NPCOMMID	0x8055160a	Invalid NP Communication ID
SCE_NP_TROPHY_ERROR_INVALID_NPCOMMSIGN	0x8055160b	Invalid NP communication signature
SCE_NP_TROPHY_ERROR_NPCOMMSIGN_VERIFICATION_FAILURE	0x8055160c	Failed to verify the NP communication signature
SCE_NP_TROPHY_ERROR_INVALID_GROUP_ID	0x8055160d	Invalid group ID
SCE_NP_TROPHY_ERROR_INVALID_TROPHY_ID	0x8055160e	Invalid trophy ID
SCE_NP_TROPHY_ERROR_TROPHY_ALREADY_UNLOCKED	0x8055160f	Trophy has already been unlocked
SCE_NP_TROPHY_ERROR_PLATINUM_CANNOT_UNLOCK	0x80551610	Platinum trophy cannot be unlocked Platinum trophies can only be unlocked by the system and cannot be unlocked by the application.
SCE_NP_TROPHY_ERROR_ACCOUNTID_NOT_MATCH	0x80551611	Account ID does not match
SCE_NP_TROPHY_ERROR_SETUP_REQUIRED	0x80551612	Context has not been set up
SCE_NP_TROPHY_ERROR_ALREADY_SETUP	0x80551613	Context has already been set up

SCE CONFIDENTIAL

Value	(Number)	Description
SCE_NP_TROPHY_ERROR_BROKEN_DATA	0x80551614	Data is corrupted
SCE_NP_TROPHY_ERROR_INSUFFICIENT_EM_SPACE	0x80551615	There is not enough available space on the internal flash memory
SCE_NP_TROPHY_ERROR_CONTEXT_ALREADY_EXISTS	0x80551616	Context already exists Multiple contexts may have been created for an NP Communication ID.
SCE_NP_TROPHY_ERROR_TRP_FILE_VERIFICATION_FAILURE	0x80551617	Failed to verify signature of the trophy pack file
SCE_NP_TROPHY_ERROR_ICON_FILE_NOT_FOUND	0x80551618	Icon file cannot be found
SCE_NP_TROPHY_ERROR_TRP_FILE_NOT_FOUND	0x80551619	Trophy pack file cannot be found
SCE_NP_TROPHY_ERROR_INVALID_TRP_FILE_FORMAT	0x8055161a	Invalid trophy pack file format
SCE_NP_TROPHY_ERROR_UNSUPPORTED_TRP_FILE	0x8055161b	Unsupported trophy pack file
SCE_NP_TROPHY_ERROR_INVALID_TROPHY_CONF_FORMAT	0x8055161c	Invalid trophy configuration
SCE_NP_TROPHY_ERROR_UNSUPPORTED_TROPHY_CONF	0x8055161d	Unsupported trophy configuration
SCE_NP_TROPHY_ERROR_TROPHY_NOT_UNLOCKED	0x8055161e	Trophy has not been unlocked
SCE_NP_TROPHY_ERROR_UNLOCK_DENIED	0x8055161f	Save data does not contain the rights to obtain trophies
SCE_NP_TROPHY_ERROR_INSUFFICIENT_MC_SPACE	0x80551620	Memory card capacity is insufficient
SCE_NP_TROPHY_ERROR_DEBUG_FAILURE	0x80551621	Debugging setting error

Return Code of Trophy Setup Dialog

Value	(Number)	Description
SCE_NP_TROPHY_SETUP_DIALOG_ERROR_PARAM	0x80102601	Invalid Trophy Setup Dialog parameters