

NP ScoreRanking Library Reference

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

Table of Contents

Datatype Definitions	4
SceNpScoreBoardId	5
SceNpScoreValue	6
SceNpScoreRankNumber	7
SceNpScorePcId	8
Datatypes	9
SceNpScoreGameInfo	10
SceNpScoreComment	11
SceNpScoreRankData	12
SceNpScorePlayerRankData	13
SceNpScoreBoardInfo	14
SceNpScoreNpIdPcId	15
SceNpScoreGetFriendRankingOptParam	16
Initialization/Termination Functions	17
sceNpScoreInit	18
sceNpScoreTerm	20
Context Operating Functions	21
sceNpScoreCreateTitleCtx	22
sceNpScoreDeleteTitleCtx	24
sceNpScoreCreateRequest	25
sceNpScoreDeleteRequest	26
sceNpScoreSetTimeout	27
sceNpScoreSetPlayerCharacterId	29
sceNpScoreChangeModeForOtherSaveDataOwners	31
sceNpScoreWaitAsync, sceNpScorePollAsync	33
Communication Processing Functions	34
sceNpScoreGetBoardInfo	35
sceNpScoreGetBoardInfoAsync	37
sceNpScoreRecordScore	38
sceNpScoreRecordScoreAsync	41
sceNpScoreRecordGameData	43
sceNpScoreRecordGameDataAsync	46
sceNpScoreGetGameData	48
sceNpScoreGetGameDataAsync	50
sceNpScoreGetRankingByNpId	52
sceNpScoreGetRankingByNpIdAsync	55
sceNpScoreGetRankingByRange	57
sceNpScoreGetRankingByRangeAsync	60
sceNpScoreCensorComment	62
sceNpScoreCensorCommentAsync	64
sceNpScoreSanitizeComment	65
sceNpScoreSanitizeCommentAsync	68
sceNpScoreGetRankingByNpIdPcId	70
sceNpScoreGetRankingByNpIdPcIdAsync	73

SCE CONFIDENTIAL

sceNpScoreGetFriendsRanking	75
sceNpScoreGetFriendsRankingAsync.....	78
sceNpScoreAbortRequest	80
Constants	81
SCE_NP_SCORE_COMMENT_MAXLEN	82
SCE_NP_SCORE_CENSOR_COMMENT_MAXLEN	83
SCE_NP_SCORE_SANITIZE_COMMENT_MAXLEN	84
SCE_NP_SCORE_GAMEINFO_MAXSIZE	85
SCE_NP_SCORE_MAX_CTX_NUM.....	86
SCE_NP_SCORE_MAX_RANGE_NUM_PER_REQUEST	87
SCE_NP_SCORE_MAX_NPID_NUM_PER_REQUEST	88
SCE_NP_SCORE_MAX_SELECTED_FRIENDS_NUM.....	89

Datatype Definitions

000004892117

SCE CONFIDENTIAL

SceNpScoreBoardId

Score board ID

Definition

```
#include <np.h>
typedef SceUInt32 SceNpScoreBoardId;
```

Description

This datatype definition is used for scoreboard IDs, to be used when specifying individual scoreboards. Scoreboards are prepared on the PSNSM ranking server per NP Communication ID. Scoreboard settings must be registered beforehand.

SCE CONFIDENTIAL

SceNpScoreValue

Score

Definition

```
#include <np.h>
typedef SceInt64 SceNpScoreValue;
```

Description

This datatype definition is used for scores. Values from -9223372036854775808 to 9223372036854775807 can be represented.

Notes

When using time, we recommend that you make settings by casting the *tick* member of *SceRtcTick*. Because the server sorts ranking by extending signs, be careful for specifications that set drastically large years/ months. Similarly, if you want to handle the range of *SceInt64*, let the title handle -9223372036854775808 as zero.

SCE CONFIDENTIAL

SceNpScoreRankNumber

Rank

Definition

```
#include <np.h>
typedef SceUInt32 SceNpScoreRankNumber;
```

Description

This datatype definition is used for ranks.

000004892117

SCE CONFIDENTIAL

SceNpScorePcId

Player character ID valid only in score ranking

Definition

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

Description

This datatype definition is used for specifying the ID when more than one score will be registered to a scoreboard from a single NP ID. 0-9 can be specified. The default is 0.

Notes

This feature can be used for example when save data can be saved at multiple moments and the scores are to be handled as scores of different players. This feature is not appropriate for handling the scores separate from the players, however. To rank the scores of game characters, for example, use a different scoreboard for each character.

Datatypes

000004892117

SceNpScoreGameInfo

Data unique to an application

Definition

```
#include <np.h>
#define SCE_NP_SCORE_GAMEINFO_MAXSIZE      (189)
typedef struct SceNpScoreGameInfo{
    SceSize infoSize;
    SceUInt8 pad[4];
    SceUInt8 data[SCE_NP_SCORE_GAMEINFO_MAXSIZE];
    SceUInt8 pad2[3];
} SceNpScoreGameInfo;
```

Members

<i>infoSize</i>	Size of the valid data in <i>data</i>
<i>pad</i>	Padding
<i>data</i>	Data area that can be freely defined for an application
<i>pad2</i>	Padding

Description

This structure is for storing data uniquely defined for an application. It can be set for all scores.
data can store other data besides character strings.

Notes

This structure cannot be used for setting and displaying text input by the user. Such text must be censored before it can be displayed. Use `SceNpScoreComment` for this purpose.

See Also

`SceNpScoreComment`

SCE CONFIDENTIAL

SceNpScoreComment

Ranking comment

Definition

```
#include <np.h>
#define SCE_NP_SCORE_COMMENT_MAXLEN (63)
typedef struct SceNpScoreComment{
    char utf8Comment [SCE_NP_SCORE_COMMENT_MAXLEN+1];
} SceNpScoreComment;
```

Members

utf8Comment Comment (UTF-8)

Description

This structure is for representing comments for scores.

Notes

The server will censor comments.

This structure can also be used for purposes other than for storing comments. For example, when using a name input by the user instead of the Online Name, this structure can be used so that the input will be censored.

SCE CONFIDENTIAL

SceNpScoreRankData

Ranking information

Definition

```
#include <np.h>
typedef struct SceNpScoreRankData{
    SceNpId npId;
    SceUInt8 reserved[49];
    SceUInt8 pad0[3];
    SceNpScorePcId pcId;
    SceNpScoreRankNumber serialRank;
    SceNpScoreRankNumber rank;
    SceNpScoreRankNumber highestRank;
    SceInt32 hasGameData;
    SceUInt8 pad1[4];
    SceNpScoreValue scoreValue;
    SceRtcTick recordDate;
} SceNpScoreRankData;
```

Members

<i>npId</i>	Structure storing NP ID of the player registering the score
<i>reserved</i>	Reserved are
<i>pad0</i>	Padding
<i>pcId</i>	Player character ID of the player registering the score. 0 is set by default
<i>serialRank</i>	Rank of the player registering the score. For same scores, the first one registered to the server will rank higher
<i>rank</i>	Rank of the player registering the score. For same scores, scores will be ranked equally
<i>highestRank</i>	Highest rank achieved by the player registering the score. <i>rank</i> value will be used
<i>hasGameData</i>	Flag to indicate whether score has game data attached or not. Only players having high ranks can record game data
<i>pad1</i>	Padding
<i>scoreValue</i>	Score of the player registering the score
<i>recordDate</i>	Time of score registration. Time of the PSN SM ranking server will be used

Description

This structure represents ranking information. Ranking information can be obtained by `sceNpScoreGetRankingByRange()` or `sceNpScoreGetRankingByRangeAsync()`.

Notes

Additional ranking information such as ranking comment and data unique to a title can be optionally obtained by `SceNpScoreComment` and `SceNpScoreGameInfo`, respectively.

SCE CONFIDENTIAL

SceNpScorePlayerRankData

Ranking information of a player

Definition

```
#include <np.h>
typedef struct SceNpScorePlayerRankData{
    SceInt32  hasData;
    SceUInt8  pad0[4];
    SceNpScoreRankData rankData;
} SceNpScorePlayerRankData;
```

Members

<i>hasData</i>	Flag indicating whether target player has rank registered or not
<i>pad0</i>	Padding
<i>rankData</i>	Ranking information of the target player. This data is only valid when <i>hasData</i> is a value other than 0

Description

This structure is for representing ranking information of a player. Ranking information can be obtained by using `sceNpScoreGetRankingByNpId()`, `sceNpScoreGetRankingByNpIdAsync()`, `sceNpScoreGetRankingByNpIdPcId()`, or `sceNpScoreGetRankingByNpIdPcIdAsync()`.

Notes

Additional ranking information such as ranking comment and data unique to a title can be optionally obtained by `SceNpScoreComment` and `SceNpScoreGameInfo`, respectively.

SCE CONFIDENTIAL

SceNpScoreBoardInfo

Scoreboard information

Definition

```
#include <np.h>
typedef struct SceNpScoreBoardInfo{
    SceUInt32 rankLimit;
    SceUInt32 updateMode;
    SceUInt32 sortMode;
    SceNpScoreRankNumber uploadNumLimit;
    SceSize uploadSizeLimit;
} SceNpScoreBoardInfo;
```

Members

<i>rankLimit</i>	Maximum number of players that can be registered to a board
<i>updateMode</i>	Rule for updating scores. Overwrite when breaking record or overwrite without exception
<i>sortMode</i>	Display order of ranked scores
<i>uploadNumLimit</i>	Minimum rank for which game data can be uploaded and attached. Players of this rank and higher can upload data
<i>uploadSizeLimit</i>	Maximum size of data that can be uploaded

For *updateMode*, one of the following values will be set.

Value	(Number)	Description
SCE_NP_SCORE_NORMAL_UPDATE	0	Overwrites when breaking own record
SCE_NP_SCORE_FORCE_UPDATE	1	Always overwrites

For *sortMode*, one of the following values will be set.

Value	(Number)	Description
SCE_NP_SCORE_DESCENDING_ORDER	0	Descending order
SCE_NP_SCORE_ASCENDING_ORDER	1	Ascending order

Description

This structure is for representing scoreboard information. It can be obtained by `sceNpScoreGetBoardInfo()` and `sceNpScoreGetBoardInfoAsync()`.

Notes

Scoreboard information is determined and set per NP Communication ID. Normally, you should not need to obtain board information from the server.

SCE CONFIDENTIAL

SceNpScoreNpIdPcId

NP ID structure with player character ID

Definition

```
#include <np.h>
typedef struct SceNpScoreNpIdPcId{
    SceNpId npId;
    SceNpScorePcId pcId;
    SceUInt8 pad[4];
} SceNpScoreNpIdPcId;
```

Members

npId NP ID
pcId Player character ID
pad Padding

Description

This structure is the NP ID with a player character ID.

SCE CONFIDENTIAL

SceNpScoreGetFriendRankingOptParam

Extended parameters for obtaining friends' scores

Definition

```
#include <np.h>
typedef struct SceNpScoreGetFriendRankingOptParam {
    SceSize size;
    SceUInt32 *startSerialRank;
    SceUInt32 *hits;
} SceNpScoreGetFriendRankingOptParam;
```

Members

<i>size</i>	Size of structure (IN)
<i>startSerialRank</i>	Pointer to start order (order among friends) to obtain (IN) If NULL is specified, obtaining will be performed starting with the first ranked friend
<i>hits</i>	Total number of friends with scores registered on the target scoreboard (OUT)

Description

This is an option structure for handling the 101th and later rankings from among friends with `sceNpScoreGetFriendsRanking()` and `sceNpScoreGetFriendsRankingAsync()`.
For *size*, specify `sizeof(SceNpScoreGetFriendRankingOptParam)`.

Notes

Web APIs are provided for preparing test environments where more than 100 friends have scores registered. For details, refer to the "Score Ranking Management Overview" document.

Initialization/Termination Functions

sceNpScoreInit

Initialize NP ScoreRanking library

Definition

```
#include <np.h>
int sceNpScoreInit(
    SceInt32 threadPriority,
    SceInt32 cpuAffinityMask,
    void *option
);
```

Arguments

<i>threadPriority</i>	Priority of the thread generated for inter-process communication
<i>cpuAffinityMask</i>	CPU affinity mask of the thread for inter-process communication
<i>option</i>	Option reserved for future extension. Always specify NULL

Return Values

Returns 0 upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_ALREADY_INITIALIZED	0x80550701	Already initialized. sceNpScoreInit() may have been executed already. Check the calling order
SCE_NP_COMMUNITY_ERROR_OUT_OF_MEMORY	0x80550703	Not enough free memory

Description

This function initializes the NP ScoreRanking library. A thread for communication with the server is created in the system process, and a thread for inter-process communication with the specified priority and affinity mask is created in the game process. Also, the necessary memory for inter-process communication will be allocated internally. This function must always be called before using the NP ScoreRanking library.

Since communication is performed in the system process, processing in the game process will be limited to inter-process communication. Since response speed also influences processing speed on the system process-side, it is recommended to set SCE_KERNEL_THREAD_CPU_AFFINITY_MASK_DEFAULT to the affinity mask and perform settings so that if a core is freed, processing is passed to it.

Examples

```
SceInt32 ret;
ret = sceNpScoreInit(
    SCE_KERNEL_DEFAULT_PRIORITY_USER,
    SCE_KERNEL_THREAD_CPU_AFFINITY_MASK_DEFAULT,
    NULL);
If (ret < 0){
    // Error handling
```

SCE CONFIDENTIAL

Note

This function is not multithread safe.

See Also

`scENpScoreTerm()`

000004892117

SCE CONFIDENTIAL

sceNpScoreTerm

Terminate NP ScoreRanking library

Definition

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

Arguments

None

Return Values

Returns 0 upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized

Description

This function terminates the NP ScoreRanking library.

Note

This function is not multithread safe.

See Also

sceNpScoreInit()

Context Operating Functions

sceNpScoreCreateTitleCtx

Create NP ScoreRanking title context

Definition

```
#include <np.h>
int sceNpScoreCreateTitleCtx(
    const SceNpCommunicationId *communicationId,
    const SceNpCommunicationPassphrase *passphrase,
    const SceNpId *selfNpId
);
```

Arguments

communicationId Pointer to the NP Communication ID.
If NULL was specified, the value of `SceNpCommunicationConfig` set with `sceNpInit()` is used

passphrase Pointer to structure storing authentication information, assigned to NP Communication ID, for score ranking.
If NULL was specified, the value of `SceNpCommunicationConfig` set with `sceNpInit()` is used

selfNpId Pointer to structure storing NP ID of logging-in user.
If NULL was specified, the NP ID obtained with `sceNpManagerGetNpId()` is used

Return Values

Returns NP ScoreRanking title context ID (>0) upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_OUT_OF_MEMORY	0x80550703	Not enough free memory
SCE_NP_COMMUNITY_ERROR_TOO_MANY_OBJECTS	0x80550706	Created more than 32 title contexts at one time. Check to see that <code>sceNpScoreDeleteTitleCtx()</code> was called as necessary

Description

This function creates an NP ScoreRanking title context to be used by NP ScoreRanking library. Normally (when a single title does not handle multiple NP Communication IDs), create a single context and use it for multiple scoreboards. You must delete a context using `sceNpScoreDeleteTitleCtx()` before calling `sceNpScoreTerm()`.

The maximum number of NP ScoreRanking title contexts that can exist at one time is 32.

Notes

Recreating the NP ScoreRanking title context will increase frequency of need for the server to obtain information for an NP Communication ID, thereby increasing its processing load. Try to make-do with 1 context. Information of the ranking server will be obtained upon first communicating with the NP ScoreRanking title context.

SCE CONFIDENTIAL

See Also

sceNpScoreDeleteTitleCtx(), SCE_NP_SCORE_MAX_CTX_NUM

000004892117

SCE CONFIDENTIAL

sceNpScoreDeleteTitleCtx

Destroy NP ScoreRanking title context

Definition

```
#include <np.h>
int sceNpScoreDeleteTitleCtx(
    SceInt32 titleCtxId
);
```

Arguments

titleCtxId ID of NP ScoreRanking title context to be deleted

Return Values

Returns 0 upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	Specified ID does not exist

Description

This function destroys the NP ScoreRanking title context used by NP ScoreRanking library.

See Also

sceNpScoreCreateTitleCtx()

SCE CONFIDENTIAL

sceNpScoreCreateRequest

Create NP ScoreRanking request ID

Definition

```
#include <np.h>
int sceNpScoreCreateRequest (
    SceInt32 titleCtxId
);
```

Arguments

titleCtxId NP ScoreRanking title context ID

Return Values

Returns NP ScoreRanking request ID (>0) upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_OUT_OF_MEMORY	0x80550703	Not enough free memory
SCE_NP_COMMUNITY_ERROR_TOO_MANY_OBJECTS	0x80550706	Created more than 32 request IDs at one time. Check to see that <code>sceNpScoreDeleteRequest()</code> was called as necessary
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	Specified ID does not exist

Description

This function creates a request ID to be used in score ranking.

This ID is used up every time communication is performed, such as score obtainment or board obtainment, and it must always be created before using these functions. Used up request IDs must be deleted with `sceNpScoreDeleteRequest()`.

The maximum number of NP ScoreRanking request IDs that can exist at one time is 32.

See Also

`sceNpScoreDeleteRequest()`, `SCE_NP_SCORE_MAX_CTX_NUM`

SCE CONFIDENTIAL

sceNpScoreDeleteRequest

Destroy NP ScoreRanking request

Definition

```
#include <np.h>
int sceNpScoreDeleteRequest(
    SceInt32 reqId
);
```

Arguments

reqId NP ScoreRanking request ID

Return Values

Returns 0 upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_OUT_OF_MEMORY	0x80550703	Not enough free memory
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	Specified ID does not exist

Description

This function destroys a request ID used in score ranking.

Transaction completion can be confirmed by the return of the function (for a synchronous function), or by `sceNpScoreWaitAsync()` or `sceNpScorePollAsync()` (for an asynchronous function).

See Also

`sceNpScoreCreateRequest()`

SCE CONFIDENTIAL

sceNpScoreSetTimeout

Set timeout for communication processing functions

Definition

```
#include <np.h>
int sceNpScoreSetTimeout(
    SceInt32 id,
    SceInt32 resolveRetry,
    SceUInt32 resolveTimeout,
    SceUInt32 connTimeout,
    SceUInt32 sendTimeout,
    SceUInt32 recvTimeout
);
```

Arguments

<i>id</i>	NP ScoreRanking title context ID or NP ScoreRanking request ID
<i>resolveRetry</i>	Name resolution retry count
<i>resolveTimeout</i>	Name resolution timeout time (in microseconds)
<i>connTimeout</i>	Timeout time when connecting (in microseconds)
<i>sendTimeout</i>	Sending timeout time (in microseconds)
<i>recvTimeout</i>	Receiving timeout time (in microseconds)

Return Values

Returns 0 upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ARGUMENT	0x80550704	Value under 10 seconds was specified to <i>connTimeout</i> , <i>sendTimeout</i> or <i>recvTimeout</i>
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	Specified ID does not exist

Description

To *id*, specify the ID of the targeted NP ScoreRanking title context or NP ScoreRanking request. Name resolution retry count and communication processing timeout time are set in microseconds. When the title context is the target, this timeout value will be used as the default value of the requests created from the title context. However, it will not affect the timeout of requests that have already been created.

Notes

In principle, do not use this function but instead use the default values. If the function will be used to prevent the user from long wait times, also implement the ability for the user to cancel the transaction using `sceNpScoreAbortRequest()`.

Default timeout values are as follows.

DNS timeout time	1 second
DNS retry count	5 times
CONNECT timeout time	30 seconds
SEND timeout time	60 seconds
RECV timeout time	60 seconds

The timeout value cannot be set in total time. This is because the waiting time of connection processing is included in the API calling time, in accordance with the newly introduced concept of intermittent connection. If processing were to be aborted when timeout occurs based on total time count, timeout would occur during time-consuming operations by the user, such as SSID and password input. Therefore, if setting timeout, it is recommended not to count on a total time basis, but rather to set socket layer timeout, as in this function.

2 seconds can be assumed to be the normal time required for communication processing. However, depending on the user's network environment, it is possible for a large delay to occur; therefore we recommend setting the timeout value to 30 seconds or more. Note that this value of 30 seconds is not appropriate for general network programming. For example, in a situation where the clients communicate directly with each other, differences in client environments and load times (depending on, for example, hardware differences), it is necessary to set a longer timeout.

See Also

`sceNpScoreCreateTitleCtx()`, `sceNpScoreCreateRequest()`,
`sceNpScoreAbortRequest()`

SCE CONFIDENTIAL

sceNpScoreSetPlayerCharacterId

Set player character ID

Definition

```
#include <np.h>
int sceNpScoreSetPlayerCharacterId(
    SceInt32 ctxId,
    SceNpScorePcId pcId
);
```

Arguments

ctxId NP ScoreRanking request ID or NP ScoreRanking title context ID
pcId Player character ID to be set
 Values between 0-9 can be set. An error will return from the server with a communication processing function if any other value is set

Return Values

Returns 0 upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ARGUMENT	0x80550704	Negative value was specified for <i>pcId</i>
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	Specified ID does not exist

Description

This function specifies the ID of an NP ScoreRanking title context (the title context) or an NP ScoreRanking request (the request) to *ctxId* and the player character ID to *pcId*.

When a title context is specified, the specified player character ID will be used as the default value of the requests created from the title context. Requests that already exist will not be affected.

The player character ID of the scores registered with `sceNpScoreRecordScore()` or `sceNpScoreRecordScoreAsync()` will be changed. If a value is not specified for the ID, 0 will be set implicitly. It is necessary to use the same player character ID for the requests when registering/getting data attachments with `sceNpScoreRecordGameData()`, `sceNpScoreRecordGameDataAsync()`, `sceNpScoreGetGameData()`, or `sceNpScoreGetGameDataAsync()`.

Notes

When a player character ID other than 0 is set, the player ranking information cannot be obtained with `sceNpScoreGetRankingByNpId()` or `sceNpScoreGetRankingByNpIdAsync()`. Note that for this reason this feature will be difficult to use alongside friend rankings. Unless there is a particular reason to have a single user register multiple scores to a scoreboard, there is no reason to use this function. For an example of its usage, refer to the description of `SceNpScorePcId`.

SCE CONFIDENTIAL

See Also

SceNpScorePcId, SceNpScoreNpIdPcId, SceNpScoreRankData,
sceNpScoreGetRankingByNpId(), sceNpScoreGetRankingByNpIdAsync(),
sceNpScoreGetRankingByNpIdPcId(), sceNpScoreGetRankingByNpIdPcIdAsync()

000004892117

SCE CONFIDENTIAL

sceNpScoreChangeModeForOtherSaveDataOwners

Set operation restriction applied when another owner's save data is loaded

Definition

```
#include <np.h>
int sceNpScoreChangeModeForOtherSaveDataOwners (
    SceInt32 ctxId,
    SceInt32 mode
);
```

Arguments

ctxId NP ScoreRanking request ID or NP ScoreRanking title context ID
mode Value representing the mode to be set

Return Values

Returns 0 upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ARGUMENT	0x80550704	Nonexistent value was specified for <i>mode</i>
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	Specified ID does not exist

Description

This function is used to prevent ranking registration using another owner's save data and prohibit the user from saving the ranking information into another owner's save data.

Since NP ScoreRanking is a system that is closed by each application, there is no TRC (Technical Requirements Checklist) requirement that prohibits the use of ranking in combination with another owner's save data. Performing a test may be difficult if using this function; thus, make sure to use this function only when it is really required according to the usage methods and policies of score ranking for an application.

For example, if the account ID written in a memory card is different from the one written in the main unit, the memory card must be formatted before it is used with the main unit.

Therefore, it is not necessary to use this function when saving the save data in a memory card. Also, this function is not required if the score is not saved in the save data. For details, refer to the "Owner Identification When Inserting Memory Cards" chapter in the "System Software Overview" document.

For *ctxId*, specify the ID of an NP ScoreRanking title context (the title context) or an NP ScoreRanking request (the request) and for *mode*, specify the operation performed when another owner's save data is loaded.

When a title context is specified, the specified value will be used as the default value of the requests created from the title context. However, requests that already exist will not be affected.

Specify one of the following values in *mode*.

mode is handled as a bit pattern not a signed integer.

Value	Description
SCE_NP_SCORE_BINDMODE_ALL_FORBIDDEN	When the save data is owned by another user, both reference from and update to the server are prohibited
SCE_NP_SCORE_BINDMODE_RDONLY	When the save data is owned by another user, reference from the server is only allowed
SCE_NP_SCORE_BINDMODE_WRONLY	When the save data is owned by another user, update to the server is only allowed
SCE_NP_SCORE_BINDMODE_RDWR	No restriction is applied even when the save data is owned by another user
SCE_NP_SCORE_BINDMODE_DEFAULT	Default setting. SCE_NP_SCORE_BINDMODE_RDWR is set

Reference APIs and update APIs are as follows.

Reference APIs

```
sceNpScoreGetBoardInfo(), sceNpScoreGetBoardInfoAsync()
sceNpScoreGetGameData(), sceNpScoreGetGameDataAsync()
sceNpScoreGetRankingByNpId(), sceNpScoreGetRankingByNpIdAsync()
sceNpScoreGetRankingByRange(), sceNpScoreGetRankingByRangeAsync()
sceNpScoreGetRankingByNpIdPcId(), sceNpScoreGetRankingByNpIdPcIdAsync()
sceNpScoreCensorComment(), sceNpScoreCensorCommentAsync()
sceNpScoreSanitizeComment(), sceNpScoreSanitizeCommentAsync()
```

Update APIs

```
sceNpScoreRecordScore(), sceNpScoreRecordScoreAsync()
sceNpScoreRecordGameData(), sceNpScoreRecordGameDataAsync()
```

When a restriction is applied, an error value

SCE_NP_COMMUNITY_ERROR_SCORE_INVALID_SAVEDATA_OWNER is received as a return value for synchronous functions or as the result obtained with `sceNpScoreWaitAsync()` or `sceNpScorePollAsync()` for asynchronous functions. No particular message is provided for this error; therefore, application should be designed to handle this error and inform the user that the application features are restricted because the save data is owned by another user.

Notes

It is possible to enable reference from and update to the ranking server irrespective of the owner of the save data by setting **Game - Fake Trophy Earning** in **★Debug Settings** of system software to **On**. Whereas this function can be used in the development environment and QA environment, it is disabled in the production environment.

For **★Debug Settings**, refer to the "★Debug Settings Functions" section of the System Software Overview" document.

See Also

```
sceNpScoreCreateTitleCtx(), sceNpScoreCreateRequest()
```


SCE CONFIDENTIAL

sceNpScoreWaitAsync, sceNpScorePollAsync

Obtain execution result of an asynchronous transaction

Definition

```
#include <np.h>
int sceNpScoreWaitAsync (
    SceInt32 reqId,
    SceInt32 *result
);
int sceNpScorePollAsync (
    SceInt32 reqId,
    SceInt32 *result
);
```

Arguments

reqId NP ScoreRanking request ID obtained with `sceNpScoreCreateRequest()`
result Pointer to variable where result of the asynchronous execution is to be stored

Return Values

Returns one of the following values upon normal termination.

(Number)	Description
0	Asynchronous processing ended
1	Asynchronous processing still being executed

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	Specified ID does not exist

Description

This function obtains execution result of an asynchronously executed transaction.

`sceNpScoreWaitAsync()` waits for the completion of the asynchronous transaction, stores this execution result in the variable pointed by *result* and returns 0.

`sceNpScorePollAsync()` is a service call in which the feature to enter a wait state - when the asynchronous function has not completed execution - has been removed from `sceNpScoreWaitAsync()`. If the asynchronous transaction is incomplete, this function returns 1. In this case, the variable pointed by *result* will not change in value.

Asynchronous transaction processing started by `sceNpScoreXXXAsync()` concludes with either `sceNpScoreWaitAsync()` or `sceNpScorePollAsync()` returning 0 and with you obtaining the result of the asynchronous transaction.

See Also

`sceNpScoreCreateRequest()`

Communication Processing Functions

SCE CONFIDENTIAL

sceNpScoreGetBoardInfo

Get scoreboard information (synchronous)

Definition

```
#include <np.h>
int sceNpScoreGetBoardInfo (
    SceInt32 reqId,
    SceNpScoreBoardId boardId,
    SceNpScoreBoardInfo *boardInfo,
    void *option
);
```

Arguments

<i>reqId</i>	NP ScoreRanking request ID
<i>boardId</i>	Target scoreboard ID (IN)
<i>boardInfo</i>	Pointer to structure where obtained board information is to be stored (OUT)
<i>option</i>	Option reserved for future extension. Always specify NULL

Return Values

Returns 0 upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ARGUMENT	0x80550704	Value other than NULL was specified to <i>option</i>
SCE_NP_COMMUNITY_ERROR_ABORTED	0x80550707	Transaction was aborted with <code>sceNpScoreAbortRequest()</code> or <code>sceNpScoreDeleteRequest()</code>
SCE_NP_COMMUNITY_ERROR_INVALID_SIGNATURE	0x8055070b	Authentication error. Usually occurs with an error in <code>SceNpCommunicationPassphrase</code>
SCE_NP_COMMUNITY_ERROR_INSUFFICIENT_ARGUMENT	0x8055070c	NULL was specified to <i>boardInfo</i>
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	ID specified for <i>reqId</i> does not exist
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_BOARD_MASTER_NOT_FOUND	0x80550817	Scoreboard specified with <i>boardId</i> does not exist
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_BEFORE_SERVICE	0x805508a3	Ranking service of the specified NP Communication ID has not yet started. Unless requested, there is no start date set for the ranking service (Corresponding message is in the Message Dialog library)

SCE CONFIDENTIAL

Value	(Number)	Description
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_END_OF_SERVICE	0x805508a4	Ranking service of the specified NP Communication ID has ended (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_MAINTENANCE	0x805508a5	Ranking service of the specified NP Communication ID is under maintenance (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_NO_SUCH_TITLE	0x805508a6	Specified NP Communication ID is not registered on the ranking server

Description

This function obtains information of the target scoreboard. The function is blocked until communication completes and scoreboard information is obtained.

See Also

`sceNpScoreCreateRequest()`, `sceNpScoreAbortRequest()`

SCE CONFIDENTIAL

sceNpScoreGetBoardInfoAsync

Get scoreboard information (asynchronous)

Definition

```
#include <np.h>
int sceNpScoreGetBoardInfoAsync (
    SceInt32 reqId,
    SceNpScoreBoardId boardId,
    SceNpScoreBoardInfo *boardInfo,
    void *option
);
```

Arguments

<i>reqId</i>	NP ScoreRanking request ID (IN)
<i>boardId</i>	Target scoreboard ID (IN)
<i>boardInfo</i>	Pointer to structure where obtained scoreboard information is to be stored (OUT)
<i>option</i>	Option reserved for future extension. Always specify NULL

Return Values

Returns 0 upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ARGUMENT	0x80550704	Value other than NULL was specified to <i>option</i>
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	Specified ID does not exist

Description

This function obtains information of the target scoreboard. Actual transaction is performed asynchronously. Get processing result using `sceNpScoreWaitAsync()`.

See Also

`sceNpScoreCreateRequest()`, `sceNpScoreAbortRequest()`, `sceNpScoreWaitAsync()`, `sceNpScorePollAsync()`

SCE CONFIDENTIAL

sceNpScoreRecordScore

Register score (synchronous)

Definition

```
#include <np.h>
int sceNpScoreRecordScore (
    SceInt32 reqId,
    SceNpScoreBoardId boardId,
    SceNpScoreValue score,
    const SceNpScoreComment *scoreComment,
    const SceNpScoreGameInfo *gameInfo,
    SceNpScoreRankNumber *tmpRank,
    const SceRtcTick *compareDate,
    void *option
);
```

Arguments

<i>reqId</i>	NP ScoreRanking request ID (IN)
<i>boardId</i>	Target scoreboard ID (IN)
<i>score</i>	Value of score to be registered (IN)
<i>scoreComment</i>	Pointer to score comment to be registered. Specify NULL if registering is not required (IN)
<i>gameInfo</i>	Pointer to data uniquely defined for a game, to be registered. Specify NULL if registering is not required (IN)
<i>tmpRank</i>	Pointer to variable where temporary rank of the registered score is to be stored (OUT).
	Specify NULL when temporary rank information is not required
<i>compareDate</i>	Specification of date/time for conflict prevention. Scores are only registered when the registration date/time of the score currently registered on the server is equal to or older than the specified date/time. When no score is registered on the server, the score is registered. Specify NULL when comparison is not required (IN)
<i>option</i>	Option reserved for future extension. Always specify NULL

Return Values

Returns 0 upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ARGUMENT	0x80550704	Value other than NULL was specified to <i>option</i> , or a value that exceeds SCE_NP_SCORE_GAMEINFO_MAXSIZE was specified to the <i>infoSize</i> member in the SceNpScoreGameInfo structure specified with <i>gameInfo</i>
SCE_NP_COMMUNITY_ERROR_ABORTED	0x80550707	Transaction was aborted with sceNpScoreAbortRequest() or sceNpScoreDeleteRequest()
SCE_NP_COMMUNITY_ERROR_INVALID_SIGNATURE	0x8055070b	Authentication error. Usually occurs with an error in SceNpCommunicationPassphrase

©SCEI

SCE CONFIDENTIAL

Value	(Number)	Description
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	ID specified for <i>reqId</i> does not exist
SCE_NP_COMMUNITY_SERVER_ERROR_FORBIDDEN	0x80550806	Specified player character ID value is invalid
SCE_NP_COMMUNITY_SERVER_ERROR_CENSORED	0x8055080b	String stored to <i>SceNpScoreComment</i> is inappropriate. Occurs when inappropriate words or a string in the form of a telephone number is included
SCE_NP_COMMUNITY_SERVER_ERROR_BLACKLISTED_USER_ID	0x80550811	Player or machine ID of PlayStation®Vita is banned from the ranking server for the specified NP Communication ID. Error does not occur when just obtaining scores (The similar macro <i>SCE_NP_COMMUNITY_SERVER_ERROR_PLAYER_BANNED</i> indicates a ban from the entire PSN SM server and does not usually occur since sign-ins will fail) (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_NOT_BEST_SCORE	0x80550815	Not the best personal score. Occurs when <i>updateMode</i> of the scoreboard is "Overwrite when breaking own record". Note that it is not always appropriate to show this error to the user. Handle this error as appropriate for the title's specifications
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_BOARD_MASTER_NOT_FOUND	0x80550817	Scoreboard specified with <i>boardId</i> does not exist
SCE_NP_COMMUNITY_SERVER_ERROR_INVALID_ANTICHEAT_DATA	0x80550819	Authentication error. Usually occurs with an error in <i>SceNpCommunicationPassphrase</i>
SCE_NP_COMMUNITY_SERVER_ERROR_NOT_RECORDABLE_VERSION	0x8055082e	Title does not have the necessary patches. Generally does not occur unless the lowest version is set to the ranking server (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_BEFORE_SERVICE	0x805508a3	Ranking service of the specified NP Communication ID has not yet started. Unless requested, there is no start date set for the ranking service (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_END_OF_SERVICE	0x805508a4	Ranking service of the specified NP Communication ID has ended (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_MAINTENANCE	0x805508a5	Ranking service of the specified NP Communication ID is under maintenance (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_NO_SUCH_TITLE	0x805508a6	Specified NP Communication ID is not registered on the ranking server

SCE CONFIDENTIAL

Description

This function registers a score to the target scoreboard and enables you to obtain the temporary rank of that score. The function is blocked until communication completes and board information is obtained.

In order to lessen server load, specify NULL to *tmpRank* when temporary rank information is not required.

Notes

Temporary rank refers to rank at the moment of the score registration. If a higher score is registered before the next official ranking sort by the server, ranking will be created with a lower rank than that of the temporary rank. Note also that when 2 players register the same score, they will be given the same temporary rank.

If the score's ranking will exceed the maximum number of registrations to the board, an error will not occur, and a rank exceeding the maximum will be stored to *tmpRank*. Scores that exceed the maximum are deleted when rankings are sorted on the server, so the value stored to *tmpRank* will never exceed the maximum greatly. When *updateMode* of the scoreboard is set to "Always overwrite", it is possible for such scores to stay within the limit if one of the upper scores goes down in the rankings during the time the server creates the rankings.

See Also

`sceNpScoreCreateRequest()`, `sceNpScoreAbortRequest()`

SCE CONFIDENTIAL

sceNpScoreRecordScoreAsync

Register score (asynchronous)

Definition

```
#include <np.h>
int sceNpScoreRecordScoreAsync (
    SceInt32 reqId,
    SceNpScoreBoardId boardId,
    SceNpScoreValue score,
    const SceNpScoreComment *scoreComment,
    const SceNpScoreGameInfo *gameInfo,
    SceNpScoreRankNumber *tmpRank,
    const SceRtcTick *compareDate,
    void *option
);
```

Arguments

<i>reqId</i>	NP ScoreRanking request ID (IN)
<i>boardId</i>	Target scoreboard ID (IN)
<i>score</i>	Value of score to be registered (IN)
<i>scoreComment</i>	Pointer to score comment to be registered (IN)
<i>gameInfo</i>	Pointer to data uniquely defined for a game, to be registered (IN)
<i>tmpRank</i>	Pointer to variable where temporary rank of the registered score is to be stored (OUT). Specify NULL when temporary rank information is not required
<i>compareDate</i>	Specification of date/time for conflict prevention. Scores are only registered when the registration date/time of the score currently registered on the server is equal to or older than the specified date/time. When no score is registered on the server, the score is registered. Specify NULL when comparison is not required (IN)
<i>option</i>	Option reserved for future extension. Always specify NULL

Return Values

Returns 0 upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ARGUMENT	0x80550704	Value other than NULL was specified to <i>option</i> , or a value that exceeds SCE_NP_SCORE_GAMEINFO_MAXSIZE was specified to the <i>infoSize</i> member in the <i>SceNpScoreGameInfo</i> structure specified with <i>gameInfo</i>
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	ID specified for <i>reqId</i> does not exist

Description

This function registers a score to the target scoreboard and enables you to obtain temporary rank of that score. Actual transaction is performed asynchronously. Get processing result using `sceNpScoreWaitAsync()`.

In order to lessen server load, specify NULL to *tmpRank* when temporary rank information is not required.

Notes

Temporary rank refers to rank at the moment of the score registration. If a higher score is registered before the next official ranking sort by the server, ranking will be created with a lower rank than that of the temporary rank. Note also that when 2 players register the same score, they will be given the same temporary rank.

See Also

`sceNpScoreCreateRequest()`, `sceNpScoreAbortRequest()`, `sceNpScoreWaitAsync()`, `sceNpScorePollAsync()`

SCE CONFIDENTIAL

sceNpScoreRecordGameData

Register game data to be attached to a score (synchronous)

Definition

```
#include <np.h>
int sceNpScoreRecordGameData (
    SceInt32 reqId,
    SceNpScoreBoardId boardId,
    SceNpScoreValue score,
    SceSize totalSize,
    SceSize sendSize,
    const void *data,
    void *option
);
```

Arguments

<i>reqId</i>	NP ScoreRanking request ID (IN)
<i>boardId</i>	Target scoreboard ID (IN)
<i>score</i>	Value of score for which game data is to be registered (IN)
<i>totalSize</i>	Total size of game data to be sent (IN)
<i>sendSize</i>	Size of game data to be sent in the current send (IN)
<i>data</i>	Pointer to game data to be sent in the current send (IN)
<i>option</i>	Option reserved for future extension. Always specify NULL

Return Values

Returns 0 upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ARGUMENT	0x80550704	Value other than NULL was specified to <i>option</i> , or 0 was specified to <i>totalSize</i>
SCE_NP_COMMUNITY_ERROR_ABORTED	0x80550707	Transaction was aborted with <code>sceNpScoreAbortRequest()</code> or <code>sceNpScoreDeleteRequest()</code>
SCE_NP_COMMUNITY_ERROR_INVALID_SIGNATURE	0x8055070b	Authentication error. Usually occurs with an error in <code>SceNpCommunicationPassphrase</code>
SCE_NP_COMMUNITY_ERROR_INSUFFICIENT_ARGUMENT	0x8055070c	NULL was specified to <i>data</i>
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	ID specified for <i>reqId</i> does not exist

SCE CONFIDENTIAL

Value	(Number)	Description
SCE_NP_COMMUNITY_SERVER_ERROR_BLACKLISTED_USER_ID	0x80550811	Player or machine ID of PlayStation®Vita is banned from the ranking server for the specified NP Communication ID. Error does not occur when just obtaining scores (The similar macro SCE_NP_COMMUNITY_SERVER_ERROR_PLAYER_BANNED indicates a ban from the entire NP and does not usually occur since sign-ins will fail) (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_STORE_NOT_FOUND	0x80550814	Player's scores could not be found
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_BOARD_MASTER_NOT_FOUND	0x80550817	Scoreboard specified with <i>boardId</i> does not exist
SCE_NP_COMMUNITY_SERVER_ERROR_TOO_LARGE_DATA	0x8055081a	Size specified to <i>totalSize</i> exceeds the maximum size set for the scoreboard
SCE_NP_COMMUNITY_SERVER_ERROR_INVALID_SCORE	0x80550823	Score is registered, but value of score is different
SCE_NP_COMMUNITY_SERVER_ERROR_GAME_DATA_ALREADY_EXISTS	0x8055082c	Specified score already has data attachment
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_BEFORE_SERVICE	0x805508a3	Ranking service of the specified NP Communication ID has not yet started. Unless requested, there is no start date set for the ranking service (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_END_OF_SERVICE	0x805508a4	Ranking service of the specified NP Communication ID has ended (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_MAINTENANCE	0x805508a5	Ranking service of the specified NP Communication ID is under maintenance (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_NO_SUCH_TITLE	0x805508a6	Specified NP Communication ID is not registered on the ranking server

Description

This function registers game data to be attached to your score on the target scoreboard. Only players of a certain rank and above can register game data to their scores. Determine eligibility from your temporary rank, which can be obtained upon score registration.

This function is blocked until communication completes and board information is obtained.

When sending game data in increments, set total game data size in *totalSize*, size of game data to be sent in the current send in *sendSize* and call this function multiple times. For *totalSize*, the value specified upon the first function call will be used throughout.

SCE CONFIDENTIAL

Notes

Temporary rank can be obtained upon score registration. When a player who is not eligible to register data sends game data to the server, that data will be destroyed from the server.

See Also

`sceNpScoreCreateRequest()`, `sceNpScoreAbortRequest()`

000004892117

SCE CONFIDENTIAL

sceNpScoreRecordGameDataAsync

Register game data to be attached to a score (asynchronous)

Definition

```
#include <np.h>
int sceNpScoreRecordGameDataAsync (
    SceInt32 reqId,
    SceNpScoreBoardId boardId,
    SceNpScoreValue score,
    SceSize totalSize,
    SceSize sendSize,
    const void *data,
    void *option
);
```

Arguments

<i>reqId</i>	NP ScoreRanking request ID (IN)
<i>boardId</i>	Target scoreboard ID (IN)
<i>score</i>	Value of score for which game data is to be registered (IN)
<i>totalSize</i>	Total size of game data to be sent (IN)
<i>sendSize</i>	Size of game data to be sent in the current send (IN)
<i>data</i>	Pointer to game data to be sent in the current send (IN)
<i>option</i>	Option reserved for future extension. Always specify NULL

Return Values

Returns 0 upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ARGUMENT	0x80550704	Value other than NULL was specified to <i>option</i>
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	ID specified for <i>reqId</i> does not exist

Description

This function registers game data to be attached to your score on the target scoreboard. Only players of a certain rank and above can register game data to their scores. Determine eligibility from your temporary rank, which can be obtained upon score registration.

Actual transaction is performed asynchronously. Get processing result using `sceNpScoreWaitAsync()`.

Refer to the section `sceNpScoreRecordGameData()` for a list of the main error codes.

When sending game data in increments, set total game data size in *totalSize*, size of game data to be sent in the current send in *sendSize* and call this function multiple times. For *totalSize*, the value specified upon the first function call will be used throughout.

SCE CONFIDENTIAL

Notes

Temporary rank can be obtained upon score registration. When a player who is not eligible to register data sends game data to the server, that data will be destroyed from the server.

Until the asynchronous processing completes, another asynchronous processing cannot be issued to the same request ID.

The memory area specified in *data* will continue to be referenced until the asynchronous transaction completes. Note that this area must be held until the result of the processing is received.

See Also

`sceNpScoreCreateRequest()`, `sceNpScoreAbortRequest()`, `sceNpScoreWaitAsync()`,
`sceNpScorePollAsync()`, `sceNpScoreRecordGameData()`

SCE CONFIDENTIAL

sceNpScoreGetGameData

Get attached game data (synchronous)

Definition

```
#include <np.h>
int sceNpScoreGetGameData (
    SceInt32 reqId,
    SceNpScoreBoardId boardId,
    const SceNpId *npId,
    SceSize *totalSize,
    SceSize recvSize,
    void *data,
    void *option
);
```

Arguments

<i>reqId</i>	NP ScoreRanking request ID (IN)
<i>boardId</i>	Target scoreboard ID (IN)
<i>npId</i>	NP ID of player who registered the target game data (IN)
<i>totalSize</i>	Pointer to memory where total size of attached game data is to be stored (IN)
<i>recvSize</i>	Size of data to be received (IN)
<i>data</i>	Pointer to memory where received game data is to be stored (OUT)
<i>option</i>	Option reserved for future extension. Always specify NULL

Return Values

Returns the received size (≥ 0) upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ARGUMENT	0x80550704	Value other than NULL was specified to <i>option</i>
SCE_NP_COMMUNITY_ERROR_ABORTED	0x80550707	Transaction was aborted with <code>sceNpScoreAbortRequest()</code> or <code>sceNpScoreDeleteRequest()</code>
SCE_NP_COMMUNITY_ERROR_INVALID_SIGNATURE	0x8055070b	Authentication error. Usually occurs with an error in <code>SceNpCommunicationPassphrase</code>
SCE_NP_COMMUNITY_ERROR_INSUFFICIENT_ARGUMENT	0x8055070c	NULL was specified to <i>npId</i>
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	ID specified for <i>reqId</i> does not exist
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_BOARD_MASTER_NOT_FOUND	0x80550817	Scoreboard specified with <i>boardId</i> does not exist
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_GAME_DATA_MASTER_NOT_FOUND	0x80550818	No data attachment exists for the specified player

SCE CONFIDENTIAL

Value	(Number)	Description
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_BEFORE_SERVICE	0x805508a3	Ranking service of the specified NP Communication ID has not yet started. Unless requested, there is no start date set for the ranking service (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_END_OF_SERVICE	0x805508a4	Ranking service of the specified NP Communication ID has ended (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_MAINTENANCE	0x805508a5	Ranking service of the specified NP Communication ID is under maintenance (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_NO_SUCH_TITLE	0x805508a6	Specified NP Communication ID is not registered on the ranking server

Description

This function receives game data attached to the score of the target player on the target scoreboard.
The function is blocked until communication completes.

When receiving game data in increments, set size of data to be received in the current receive in *recvSize* and call this function multiple times. Total size of attached game data can be obtained upon the first function call. You can specify a value for *recvSize* which exceeds the total size of data to be received.

Notes

If attached game data is updated while receiving data, you will continue to receive game data that you started to receive upon first calling this function until the end. Size of data cannot suddenly be changed, and data - before and after the update - cannot be mixed up.

See Also

`sceNpScoreCreateRequest()`, `sceNpScoreAbortRequest()`

SCE CONFIDENTIAL

sceNpScoreGetGameDataAsync

Get attached game data (asynchronous)

Definition

```
#include <np.h>
int sceNpScoreGetGameDataAsync (
    SceInt32 reqId,
    SceNpScoreBoardId boardId,
    const SceNpId *npId,
    SceSize *totalSize,
    SceSize recvSize,
    void *data,
    void *option
);
```

Arguments

<i>reqId</i>	NP ScoreRanking request ID (IN)
<i>boardId</i>	Target scoreboard ID (IN)
<i>npId</i>	NP ID of player who registered the target game data (IN)
<i>totalSize</i>	Pointer to memory where total size of attached game data is to be stored (IN)
<i>recvSize</i>	Size of data to be received (IN)
<i>data</i>	Pointer to memory where received game data is to be stored (OUT)
<i>option</i>	Option reserved for future extension. Always specify NULL

Return Values

Returns 0 upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ARGUMENT	0x80550704	Value other than NULL was specified to <i>option</i>
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	ID specified for <i>reqId</i> does not exist

Description

This function receives game data attached to the score of the target player on the target scoreboard. Actual transaction is performed asynchronously. Get processing result using `sceNpScoreWaitAsync()`.

Refer to the section `sceNpScoreGetGameData()` for a list of the main error codes.

When receiving game data in increments, set size of data to be received in the current receive in *recvSize* and call this function multiple times. Total size of attached game data can be obtained upon the first function call. You can specify a value for *recvSize* which exceeds the total size of data to be received.

SCE CONFIDENTIAL

Notes

Until the asynchronous processing completes, another asynchronous processing cannot be issued to the same request ID.

If attached game data is updated while receiving data, you will continue to receive game data that you started to receive upon first calling this function until the end. Size of data cannot suddenly be changed, and data - before and after the update, cannot be mixed up.

See Also

`sceNpScoreCreateRequest()`, `sceNpScoreAbortRequest()`, `sceNpScoreWaitAsync()`,
`sceNpScorePollAsync()`, `sceNpScoreGetGameData()`

000004892117

SCE CONFIDENTIAL

sceNpScoreGetRankingByNpId

Get ranking information of target players (synchronous)

Definition

```
#include <np.h>
int sceNpScoreGetRankingByNpId(
    SceInt32 reqId,
    SceNpScoreBoardId boardId,
    const SceNpId *npIdArray,
    SceSize npIdArraySize,
    SceNpScorePlayerRankData *rankArray,
    SceSize rankArraySize,
    SceNpScoreComment *commentArray,
    SceSize commentArraySize,
    SceNpScoreGameInfo *infoArray,
    SceSize infoArraySize,
    SceSize arrayNum,
    SceRtcTick *lastSortDate,
    SceNpScoreRankNumber *totalRecord,
    void *option
);
```

Arguments

<i>reqId</i>	NP ScoreRanking request ID (IN)
<i>boardId</i>	Target scoreboard ID (IN)
<i>npIdArray</i>	Pointer to top array of structure where NP IDs of target players are to be stored. This area must be 8-byte aligned and continuous (IN)
<i>npIdArraySize</i>	Memory size of <i>npIdArray</i> . Specify (<i>arrayNum</i> * sizeof(<i>SceNpId</i>)) (IN)
<i>rankArray</i>	Pointer to top array of structure where ranking information is to be stored. This area must be 8-byte aligned and continuous (OUT)
<i>rankArraySize</i>	Memory size of <i>rankArray</i> . Specify (<i>arrayNum</i> * sizeof(<i>SceNpScorePlayerRankData</i>)) (IN)
<i>commentArray</i>	Pointer to top array of structure where NP ScoreRanking comments are to be stored. This area must be 8-byte aligned and continuous. Comments will not be set when NULL is specified here (OUT)
<i>commentArraySize</i>	Memory size of <i>commentArray</i> . Specify (<i>arrayNum</i> * sizeof(<i>SceNpScoreComment</i>)). This argument will not be referenced when NULL is specified in <i>commentArray</i> (IN)
<i>infoArray</i>	Pointer to top array of structure where data unique to a title is to be stored. This area must be 8-byte aligned and continuous. Data will not be set when NULL is specified here (OUT)
<i>infoArraySize</i>	Memory size of <i>infoArray</i> . Specify (<i>arrayNum</i> * sizeof(<i>SceNpScoreGameInfo</i>)). This argument will not be referenced when NULL is specified in <i>infoArray</i> (IN)
<i>arrayNum</i>	Number of arrays in the specified <i>npIdArray</i> . Maximum number is 101 players (IN)
<i>lastSortDate</i>	Time when server created ranking information (UTC) (OUT)
<i>totalRecord</i>	Total number of players registered in the target scoreboard (OUT)
<i>option</i>	Option reserved for future extension. Always specify NULL

SCE CONFIDENTIAL

Return Values

Returns the number of obtained ranking information - in other words, the number of *hasData* members with positive values (>0) out of all the structure arrays of *rankArray* for normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ARGUMENT	0x80550704	Value other than NULL was specified to <i>option</i>
SCE_NP_COMMUNITY_ERROR_ABORTED	0x80550707	Transaction was aborted with <i>sceNpScoreAbortRequest()</i> or <i>sceNpScoreDeleteRequest()</i>
SCE_NP_COMMUNITY_ERROR_INVALID_SIGNATURE	0x8055070b	Authentication error. Usually occurs with an error in <i>SceNpCommunicationPassphrase</i>
SCE_NP_COMMUNITY_ERROR_INSUFFICIENT_ARGUMENT	0x8055070c	NULL was specified to <i>npIdArray</i> , <i>rankArray</i> , <i>totalRecord</i> , or <i>lastSortDate</i> . Or 0 was specified to <i>arrayNum</i>
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	ID specified for <i>reqId</i> does not exist
SCE_NP_COMMUNITY_ERROR_INVALID_ALIGNMENT	0x80550714	Size of structure specified to <i>rankArraySize</i> , <i>commentArraySize</i> , or <i>infoArraySize</i> is invalid
SCE_NP_COMMUNITY_ERROR_TOO_MANY_NPID	0x80550719	Value exceeding 101 was specified to <i>arrayNum</i>
SCE_NP_COMMUNITY_SERVER_ERROR_GAME_RANKING_NOT_FOUND	0x80550812	None of the specified players has scores registered
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_BOARD_MASTER_NOT_FOUND	0x80550817	Scoreboard specified with <i>boardId</i> does not exist
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_BEFORE_SERVICE	0x805508a3	Ranking service of the specified NP Communication ID has not yet started. Unless requested, there is no start date set for the ranking service (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_END_OF_SERVICE	0x805508a4	Ranking service of the specified NP Communication ID has ended (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_MAINTENANCE	0x805508a5	Ranking service of the specified NP Communication ID is under maintenance (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_NO_SUCH_TITLE	0x805508a6	Specified NP Communication ID is not registered on the ranking server

Description

This function obtains ranking information of target players on the target scoreboard. The function is blocked until communication completes and board information is obtained.

SCE CONFIDENTIAL

See Also

`sceNpScoreCreateRequest()`, `sceNpScoreAbortRequest()`

000004892117

SCE CONFIDENTIAL

sceNpScoreGetRankingByNpIdAsync

Get ranking information of target players (asynchronous)

Definition

```
#include <np.h>
int sceNpScoreGetRankingByNpIdAsync (
    SceInt32 reqId,
    SceNpScoreBoardId boardId,
    const SceNpId *npIdArray,
    SceSize npIdArraySize,
    SceNpScorePlayerRankData *rankArray,
    SceSize rankArraySize,
    SceNpScoreComment *commentArray,
    SceSize commentArraySize,
    SceNpScoreGameInfo *infoArray,
    SceSize infoArraySize,
    SceSize arrayNum,
    SceRtcTick *lastSortDate,
    SceNpScoreRankNumber *totalRecord,
    void *option
);
```

Arguments

<i>reqId</i>	NP ScoreRanking request ID (IN)
<i>boardId</i>	Target scoreboard ID (IN)
<i>npIdArray</i>	Pointer to top array of structure where NP IDs of target players are to be stored. This area must be 8-byte aligned and continuous (IN)
<i>npIdArraySize</i>	Memory size of <i>npIdArray</i> . Specify (<i>arrayNum</i> * sizeof(SceNpId)) (IN)
<i>rankArray</i>	Pointer to top array of structure where ranking information is to be stored. This area must be 8-byte aligned and continuous (OUT)
<i>rankArraySize</i>	Memory size of <i>rankArray</i> . Specify (<i>arrayNum</i> * sizeof(SceNpScorePlayerRankData)) (IN)
<i>commentArray</i>	Pointer to top array of structure where NP ScoreRanking comments are to be stored. This area must be 8-byte aligned and continuous. Comment will not be set when NULL is specified here (OUT)
<i>commentArraySize</i>	Memory size of <i>commentArray</i> . Specify (<i>arrayNum</i> * sizeof(SceNpScoreComment)). This argument will not be referenced when NULL is specified in <i>commentArray</i> (IN)
<i>infoArray</i>	Pointer to top array of structure where data unique to a title is to be stored. This area must be 8-byte aligned and continuous. Data will not be set when NULL is specified here (OUT)
<i>infoArraySize</i>	Memory size of <i>infoArray</i> . Specify (<i>arrayNum</i> * sizeof(SceNpScoreGameInfo)). This argument will not be referenced when NULL is specified in <i>infoArray</i> (IN)
<i>arrayNum</i>	Number of arrays in the specified <i>npIdArray</i> . Maximum number is 101 players (IN)
<i>lastSortDate</i>	Time when server created ranking information (UTC) (OUT)
<i>totalRecord</i>	Total number of players registered in the target scoreboard (OUT)
<i>option</i>	Option reserved for future extension. Always specify NULL

©SCEI

SCE CONFIDENTIAL

Return Values

Returns 0 upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ARGUMENT	0x80550704	Value other than NULL was specified to <i>option</i>
SCE_NP_COMMUNITY_ERROR_INSUFFICIENT_ARGUMENT	0x8055070c	NULL was specified to <i>npIdArray</i> . Or 0 was specified to <i>arrayNum</i>
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	ID specified for <i>reqId</i> does not exist

Description

This function obtains ranking information of target players on the target scoreboard. Actual transaction is performed asynchronously. Get processing result using `sceNpScoreWaitAsync()`.

Refer to the section `sceNpScoreGetRankingByNpId()` for a list of the main error codes.

See Also

`sceNpScoreCreateRequest()`, `sceNpScoreAbortRequest()`, `sceNpScoreWaitAsync()`, `sceNpScorePollAsync()`, `sceNpScoreGetRankingByNpId()`

SCE CONFIDENTIAL

sceNpScoreGetRankingByRange

Get ranking information beginning with the specified rank (synchronous)

Definition

```
#include <np.h>
int sceNpScoreGetRankingByRange (
    SceInt32 reqId,
    SceNpScoreBoardId boardId,
    SceNpScoreRankNumber startSerialRank,
    SceNpScoreRankData *rankArray,
    SceSize rankArraySize,
    SceNpScoreComment *commentArray,
    SceSize commentArraySize,
    SceNpScoreGameInfo *infoArray,
    SceSize infoArraySize,
    SceSize arrayNum,
    SceRtcTick *lastSortDate,
    SceNpScoreRankNumber *totalRecord,
    void *option
);
```

Arguments

<i>reqId</i>	NP ScoreRanking request ID (IN)
<i>boardId</i>	Target scoreboard ID (IN)
<i>startSerialRank</i>	Beginning position where ranking data to be obtained is to start. Can only be specified for serial ranking (IN)
<i>rankArray</i>	Pointer to top array of structure where ranking information is to be stored. This area must be 8-byte aligned and continuous (OUT)
<i>rankArraySize</i>	Memory size of <i>rankArray</i> . Specify (<i>arrayNum</i> * sizeof(SceNpScoreRankData)) (IN)
<i>commentArray</i>	Pointer to top array of structure where NP ScoreRanking comment is to be stored. This area must be 8-byte aligned and continuous. Comment will not be set when NULL is specified here (OUT)
<i>commentArraySize</i>	Memory size of <i>commentArray</i> . Specify (<i>arrayNum</i> * sizeof(SceNpScoreComment)). This argument will not be referenced when NULL is specified in <i>commentArray</i> (IN)
<i>infoArray</i>	Pointer to top array of structure where data unique to a title is to be stored. This area must be 8-byte aligned and continuous. Data will not be set when NULL is specified here (OUT)
<i>infoArraySize</i>	Memory size of <i>infoArray</i> . Specify (<i>arrayNum</i> * sizeof(SceNpScoreGameInfo)). This argument will not be referenced when NULL is specified in <i>infoArray</i> (IN)
<i>arrayNum</i>	Number of ranking information to obtain. Maximum number is 100 (IN)
<i>lastSortDate</i>	Time when server created ranking information (UTC) (OUT)
<i>totalRecord</i>	Total number of players registered on the target scoreboard (OUT)
<i>option</i>	Option reserved for future extension. Always specify NULL

SCE CONFIDENTIAL

Return Values

Returns the number of ranking information obtained (>0) upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ARGUMENT	0x80550704	One of the following - 0 was specified to <i>startSerialRank</i> - Value exceeding 100 was specified to <i>arrayNum</i> - value other than NULL was specified to <i>option</i>
SCE_NP_COMMUNITY_ERROR_ABORTED	0x80550707	Transaction was aborted with <i>sceNpScoreAbortRequest()</i> or <i>sceNpScoreDeleteRequest()</i>
SCE_NP_COMMUNITY_ERROR_INVALID_SIGNATURE	0x8055070b	Authentication error. Usually occurs with an error in <i>SceNpCommunicationPassphrase</i>
SCE_NP_COMMUNITY_ERROR_INSUFFICIENT_ARGUMENT	0x8055070c	NULL was specified to <i>rankArray</i> , <i>totalRecord</i> , or <i>lastSortDate</i> . Or 0 was specified to <i>arrayNum</i>
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	ID specified for <i>reqId</i> does not exist
SCE_NP_COMMUNITY_ERROR_INVALID_ALIGNMENT	0x80550714	Size of structure specified to <i>rankArraySize</i> , <i>commentArraySize</i> , or <i>infoArraySize</i> is invalid
SCE_NP_COMMUNITY_SERVER_ERROR_GAME_RANKING_NOT_FOUND	0x80550812	No scores of the specified conditions could be found
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_BOARD_MASTER_NOT_FOUND	0x80550817	Scoreboard specified with <i>boardId</i> does not exist
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_BEFORE_SERVICE	0x805508a3	Ranking service of the specified NP Communication ID has not yet started. Unless requested, there is no start date set for the ranking service (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_END_OF_SERVICE	0x805508a4	Ranking service of the specified NP Communication ID has ended (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_MAINTENANCE	0x805508a5	Ranking service of the specified NP Communication ID is under maintenance (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_NO_SUCH_TITLE	0x805508a6	Specified NP Communication ID is not registered on the ranking server

Description

This function obtains ranking information of the target scoreboard. The function is blocked until communication completes and board information is obtained.

The value that can be specified to *startSerialRank* is 1 or more. When 0 is specified, this function returns SCE_NP_COMMUNITY_ERROR_INVALID_ARGUMENT.

SCE CONFIDENTIAL

See Also

`sceNpScoreCreateRequest()`, `sceNpScoreAbortRequest()`

000004892117

sceNpScoreGetRankingByRangeAsync

Get ranking information beginning with the specified rank (asynchronous)

Definition

```
#include <np.h>
int sceNpScoreGetRankingByRangeAsync (
    SceInt32 reqId,
    SceNpScoreBoardId boardId,
    SceNpScoreRankNumber startSerialRank,
    SceNpScoreRankData *rankArray,
    SceSize rankArraySize,
    SceNpScoreComment *commentArray,
    SceSize commentArraySize,
    SceNpScoreGameInfo *infoArray,
    SceSize infoArraySize,
    SceSize arrayNum,
    SceRtcTick *lastSortDate,
    SceNpScoreRankNumber *totalRecord,
    void *option
);
```

Arguments

<i>reqId</i>	NP ScoreRanking request ID (IN)
<i>boardId</i>	Target scoreboard ID (IN)
<i>startSerialRank</i>	Beginning position where ranking data to be obtained is to start. Can only be specified for serial ranking (IN)
<i>rankArray</i>	Pointer to top array of structure where ranking information is to be stored. This area must be 8-byte aligned and continuous (OUT)
<i>rankArraySize</i>	Memory size of <i>rankArray</i> . Specify (<i>arrayNum</i> * sizeof(<i>SceNpScoreRankData</i>)) (IN)
<i>commentArray</i>	Pointer to top array of structure where NP ScoreRanking comment is to be stored. This area must be 8-byte aligned and continuous. Comment will not be set when NULL is specified here (OUT)
<i>commentArraySize</i>	Memory size of <i>commentArray</i> . Specify (<i>arrayNum</i> * sizeof(<i>SceNpScoreComment</i>)). This argument will not be referenced when NULL is specified in <i>commentArray</i> (IN)
<i>infoArray</i>	Pointer to top array of structure where data unique to a title is to be stored. This area must be 8-byte aligned and continuous. Data will not be set when NULL is specified here (OUT)
<i>infoArraySize</i>	Memory size of <i>infoArray</i> . Specify (<i>arrayNum</i> * sizeof(<i>SceNpScoreGameInfo</i>)). This argument will not be referenced when NULL is specified in <i>infoArray</i> (IN)
<i>arrayNum</i>	Number of ranking information to obtain. Maximum number is 100 (IN)
<i>lastSortDate</i>	Time when server created ranking information (UTC) (OUT)
<i>totalRecord</i>	Total number of players registered on the target scoreboard (OUT)
<i>option</i>	Option reserved for future extension. Always specify NULL

SCE CONFIDENTIAL

Return Values

Returns 0 upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ARGUMENT	0x80550704	0 was specified to <i>startSerialRank</i> or value other than NULL was specified to <i>option</i>
SCE_NP_COMMUNITY_ERROR_INSUFFICIENT_ARGUMENT	0x8055070c	0 was specified to <i>arrayNum</i>
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	ID specified for <i>reqId</i> does not exist

Description

This function obtains ranking information of the target scoreboard. Actual transaction is performed asynchronously. Get processing result using `sceNpScoreWaitAsync()`.

The value that can be specified to *startSerialRank* is 1 or more. When 0 is specified, this function returns SCE_NP_COMMUNITY_ERROR_INVALID_ARGUMENT.

See Also

`sceNpScoreCreateRequest()`, `sceNpScoreAbortRequest()`, `sceNpScoreWaitAsync()`, `sceNpScorePollAsync()`

SCE CONFIDENTIAL

sceNpScoreCensorComment

Check for inappropriate character strings (synchronous)

Definition

```
#include <np.h>
int sceNpScoreCensorComment(
    SceInt32 reqId,
    const char *comment,
    void *option
);
```

Arguments

reqId NP ScoreRanking request ID (IN)
comment Pointer to UTF-8 character string to check; maximum 255 bytes (IN)
option Option reserved for future extension. Always specify NULL

Return Values

Returns 0 upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ARGUMENT	0x80550704	UTF-8 character string of 256 bytes or more was specified to <i>comment</i> , or value other than NULL was specified to <i>option</i>
SCE_NP_COMMUNITY_ERROR_ABORTED	0x80550707	Transaction was aborted with <code>sceNpScoreAbortRequest()</code> or <code>sceNpScoreDeleteRequest()</code>
SCE_NP_COMMUNITY_ERROR_INVALID_SIGNATURE	0x8055070b	Authentication error. Usually occurs with an error in <code>SceNpCommunicationPassphrase</code>
SCE_NP_COMMUNITY_ERROR_INSUFFICIENT_ARGUMENT	0x8055070c	NULL was specified to <i>comment</i>
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	ID specified for <i>reqId</i> does not exist
SCE_NP_COMMUNITY_SERVER_ERROR_CENSORED	0x8055080b	String specified to <i>comment</i> is inappropriate. Occurs when inappropriate words or a string in the form of a telephone number is included
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_BEFORE_SERVICE	0x805508a3	Ranking service of the specified NP Communication ID has not yet started. Unless requested, there is no start date set for the ranking service (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_END_OF_SERVICE	0x805508a4	Ranking service of the specified NP Communication ID has ended (Corresponding message is in the Message Dialog library)

SCE CONFIDENTIAL

Value	(Number)	Description
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_MAINTENANCE	0x805508a5	Ranking service of the specified NP Communication ID is under maintenance (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_NO_SUCH_TITLE	0x805508a6	Specified NP Communication ID is not registered on the ranking server

Description

This function is the comment censoring feature of `sceNpScoreRecordScore()`. The function is blocked until communication completes and scoreboard information is obtained.

Notes

It is not necessary to check strings used in chatting, for example. Use this function to check strings that will be visible to all players and will remain visible over a long time. Note that the rules regarding inappropriate words are subject to change without notice.

As a library that is exclusively used for censoring comments, the NP WordFilter library is provided. If the ranking feature is not required, using the NP WordFilter library enables the resource consumption to be reduced by the size of PRX, etc.

See Also

`sceNpScoreCreateRequest()`, `sceNpScoreAbortRequest()`,
`SCE_NP_SCORE_CENSOR_COMMENT_MAXLEN`

SCE CONFIDENTIAL

sceNpScoreCensorCommentAsync

Check for inappropriate character strings (asynchronous)

Definition

```
#include <np.h>
int sceNpScoreCensorCommentAsync (
    SceInt32 reqId,
    const char *comment,
    void *option
);
```

Arguments

<i>reqId</i>	NP ScoreRanking request ID (IN)
<i>comment</i>	Pointer to UTF-8 character string to check; maximum 255 bytes (IN)
<i>option</i>	Option reserved for future extension. Always specify NULL

Return Values

Returns 0 upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ARGUMENT	0x80550704	UTF-8 character string of 256 bytes or more was specified to <i>comment</i> , or value other than NULL was specified to <i>option</i>
SCE_NP_COMMUNITY_ERROR_INSUFFICIENT_ARGUMENT	0x8055070c	NULL was specified to <i>comment</i>
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	ID specified for <i>reqId</i> does not exist

Description

This function is the comment censoring feature of `sceNpScoreRecordScoreAsync()`. The actual transaction is executed asynchronously and to receive the result, a function such as `sceNpScoreWaitAsync()` must always be used.

Notes

It is not necessary to check strings used in chatting, for example. Use this function to check strings that will be visible to all players and will remain visible over a long time. Note that the rules regarding inappropriate words are subject to change without notice.

As a library that is exclusively used for censoring comments, the NP WordFilter library is provided. If the ranking feature is not required, using the NP WordFilter library enables the resource consumption to be reduced by the size of PRX, etc.

See Also

`sceNpScoreCreateRequest()`, `sceNpScoreAbortRequest()`, `sceNpScoreWaitAsync()`, `sceNpScorePollAsync()`, `SCE_NP_SCORE_CENSOR_COMMENT_MAXLEN`

SCE CONFIDENTIAL

sceNpScoreSanitizeComment

Correct inappropriate character string (synchronous)

Definition

```
#include <np.h>
int sceNpScoreSanitizeComment(
    SceInt32 reqId,
    const char *comment,
    char *sanitizedComment,
    void *option
);
```

Arguments

<i>reqId</i>	NP ScoreRanking request ID (IN)
<i>comment</i>	Pointer to UTF-8 character string to correct; maximum 255 bytes (IN)
<i>sanitizedComment</i>	Address of memory for storing the UTF-8 character string after correction. Prepare a size of <code>strlen(comment)+1</code> (OUT)
<i>option</i>	Option reserved for future extension. Always specify NULL

Return Values

Returns 0 upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ARGUMENT	0x80550704	UTF-8 character string of 256 bytes or more was specified to <i>comment</i> , or value other than NULL was specified to <i>option</i>
SCE_NP_COMMUNITY_ERROR_ABORTED	0x80550707	Transaction was aborted with <code>sceNpScoreAbortRequest()</code> or <code>sceNpScoreDeleteRequest()</code>
SCE_NP_COMMUNITY_ERROR_INVALID_SIGNATURE	0x8055070b	Authentication error. Usually occurs with an error in <code>SceNpCommunicationPassphrase</code>
SCE_NP_COMMUNITY_ERROR_INSUFFICIENT_ARGUMENT	0x8055070c	NULL was specified to <i>comment</i>
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	ID specified for <i>reqId</i> does not exist
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_BEFORE_SERVICE	0x805508a3	Ranking service of the specified NP Communication ID has not yet started. Unless requested, there is no start date set for the ranking service (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_END_OF_SERVICE	0x805508a4	Ranking service of the specified NP Communication ID has ended (Corresponding message is in the Message Dialog library)

SCE CONFIDENTIAL

Value	(Number)	Description
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_MAINTENANCE	0x805508a5	Ranking service of the specified NP Communication ID is under maintenance (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_NO_SUCH_TITLE	0x805508a6	Specified NP Communication ID is not registered on the ranking server

Description

While `sceNpScoreCensorComment()` returns `SCE_NP_COMMUNITY_SERVER_ERROR_CENSORED` upon identifying an inappropriate comment, this function converts the inappropriate characters to '*' and stores the converted string in `sanitizedComment`. Even if an inappropriate character string is included, the processing returns 0. To evaluate whether an inappropriate character string was included or not can only be confirmed by comparing whether the character string specified in `comment` is the same as what is stored in `sanitizedComment`.

Because the multi-byte character of UTF-8 can be converted to '*'(0x2a), the number of bytes for the UTF-8 character string stored in `sanitizedComment` can become smaller than the byte size of the UTF-8 character string specified in `comment`.

The function is blocked until communication completes and scoreboard information is obtained.

Examples

```

SceInt32 ret;
SceInt32 reqId, titleCtxId;
char *comment="target comment\n xxx\n";
char *sanitizedComment=NULL;
SceSize commentSize;

// Assuming appropriate values are stored in reqId and titleCtxId:

commentSize = strlen(comment)+1;
sanitizedComment = (char*)malloc(commentSize);
if (sanitizedComment == NULL){
    printf("Can't alloc memory for comment\n");
    goto error;
}
memset(sanitizedComment, 0x00,commentSize);

ret = sceNpScoreSanitizeComment(reqId, comment, sanitizedComment,NULL);
if (ret < 0) {
    printf("sceNpScoreSanitizeComment() failed. ret = 0x%x\n", ret);
    goto error;
}
printf("=====\n");
printf("--original comment\n");
printf("%s\n", comment);
if (strcmp(comment, sanitizedComment) == 0){
    printf("%s may be proper\n", comment);
} else {
    printf("--sanitized comment\n");
    printf("%s\n", sanitizedComment);
}

```

SCE CONFIDENTIAL

Notes

It is not necessary to check strings used in chatting, for example. Use this function to check strings that will be visible to all players and will remain visible over a long time. Note that the rules regarding inappropriate words are subject to change without notice.

As a library that is exclusively used for censoring comments, the NP WordFilter library is provided. If the ranking feature is not required, using the NP WordFilter library enables the resource consumption to be reduced by the size of PRX, etc.

See Also

```
sceNpScoreCreateRequest(), sceNpScoreAbortRequest(),  
SCE_NP_SCORE_SANITIZE_COMMENT_MAXLEN
```

000004892117

SCE CONFIDENTIAL

sceNpScoreSanitizeCommentAsync

Correct inappropriate character string (asynchronous)

Definition

```
#include <np.h>
int sceNpScoreSanitizeCommentAsync (
    SceInt32 reqId,
    const char *comment,
    char *sanitizedComment,
    void *option
);
```

Arguments

<i>reqId</i>	NP ScoreRanking request ID (IN)
<i>comment</i>	Pointer to UTF-8 character string to correct; maximum 255 bytes (IN)
<i>sanitizedComment</i>	Address of memory for storing the UTF-8 character string after correction. Prepare a size of <code>strlen(comment)+1</code> (OUT)
<i>option</i>	Option reserved for future extension. Always specify NULL

Return Values

Returns 0 upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ARGUMENT	0x80550704	UTF-8 character string of 256 bytes or more was specified to <i>comment</i> , or value other than NULL was specified to <i>option</i>
SCE_NP_COMMUNITY_ERROR_INSUFFICIENT_ARGUMENT	0x8055070c	NULL was specified to <i>comment</i>
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	ID specified for <i>reqId</i> does not exist

Description

While `sceNpScoreCensorCommentAsync()` returns

`SCE_NP_COMMUNITY_SERVER_ERROR_CENSORED` upon identifying an inappropriate comment, this function converts the inappropriate characters to '*' and stores the converted string in *sanitizedComment*. Even if an inappropriate character string is included, the processing returns 0. To evaluate whether an inappropriate character string was included or not can only be confirmed by comparing whether the character string specified in *comment* is the same as what is stored in *sanitizedComment*.

Because the multi-byte character of UTF-8 can be converted to '*'(0x2a), the number of bytes for the UTF-8 character string stored in *sanitizedComment* can become smaller than the byte size of the UTF-8 character string specified in *comment*.

The actual transaction is executed asynchronously and to receive the result, a function such as `sceNpScoreWaitAsync()` must always be used.

SCE CONFIDENTIAL

Notes

It is not necessary to check strings used in chatting, for example. Use this function to check strings that will be visible to all players and will remain visible over a long time. Note that the rules regarding inappropriate words are subject to change without notice.

As a library that is exclusively used for censoring comments, the NP WordFilter library is provided. If the ranking feature is not required, using the NP WordFilter library enables the resource consumption to be reduced by the size of PRX, etc.

See Also

`sceNpScoreCreateRequest()`, `sceNpScoreAbortRequest()`, `sceNpScoreWaitAsync()`,
`sceNpScorePollAsync()`, `SCE_NP_SCORE_SANITIZE_COMMENT_MAXLEN`

SCE CONFIDENTIAL

sceNpScoreGetRankingByNpIdPcId

Get ranking information of player with player character ID specified (synchronous)

Definition

```
#include <np.h>
int sceNpScoreGetRankingByNpIdPcId(
    SceInt32 reqId,
    SceNpScoreBoardId boardId,
    const SceNpScoreNpIdPcId *idArray,
    SceSize idArraySize,
    SceNpScorePlayerRankData *rankArray,
    SceSize rankArraySize,
    SceNpScoreComment *commentArray,
    SceSize commentArraySize,
    SceNpScoreGameInfo *infoArray,
    SceSize infoArraySize,
    SceSize arrayNum,
    SceRtcTick *lastSortDate,
    SceNpScoreRankNumber *totalRecord,
    void *option
);
```

Arguments

<i>reqId</i>	NP ScoreRanking request ID (IN)
<i>boardId</i>	Target scoreboard ID (IN)
<i>idArray</i>	Pointer to top array of the target player NP ID structure with player character ID. This area must be 8-byte aligned and continuous (IN)
<i>idArraySize</i>	Memory size of <i>idArray</i> . Specify (<i>arrayNum</i> * sizeof(SceNpScoreNpIdPcId)) (IN)
<i>rankArray</i>	Pointer to top array of structure where ranking information is to be stored. This area must be 8-byte aligned and continuous (OUT)
<i>rankArraySize</i>	Memory size of <i>rankArray</i> . Specify (<i>arrayNum</i> * sizeof(SceNpScorePlayerRankData)) (IN)
<i>commentArray</i>	Pointer to top array of structure where NP ScoreRanking comment is to be stored. This area must be 8-byte aligned and continuous. Comment will not be set when NULL is specified here (OUT)
<i>commentArraySize</i>	Memory size of <i>commentArray</i> . Specify (<i>arrayNum</i> * sizeof(SceNpScoreComment)). This argument will not be referenced when NULL is specified in <i>commentArray</i> (IN)
<i>infoArray</i>	Pointer to top array of structure where data unique to a title is to be stored. This area must be 8-byte aligned and continuous. Data will not be set when NULL is specified here (OUT)
<i>infoArraySize</i>	Memory size of <i>infoArray</i> . Specify (<i>arrayNum</i> * sizeof(SceNpScoreGameInfo)). This argument will not be referenced when NULL is specified in <i>infoArray</i> (IN)
<i>arrayNum</i>	Number of arrays in <i>idArray</i> . Maximum number is 101 (IN).
<i>lastSortDate</i>	Time when server created ranking information (UTC) (OUT)
<i>totalRecord</i>	Total number of players registered on the target scoreboard (OUT)
<i>option</i>	Option reserved for future extension. Always specify NULL

SCE CONFIDENTIAL

Return Values

Returns the number of obtained ranking information - in other words, the number of *hasData* members with positive values (>0) out of all the structure arrays of *rankArray* for normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ARGUMENT	0x80550704	Value other than NULL was specified to <i>option</i>
SCE_NP_COMMUNITY_ERROR_ABORTED	0x80550707	Transaction was aborted with <i>sceNpScoreAbortRequest()</i> or <i>sceNpScoreDeleteRequest()</i>
SCE_NP_COMMUNITY_ERROR_INVALID_SIGNATURE	0x8055070b	Authentication error. Usually occurs with an error in <i>SceNpCommunicationPassphrase</i>
SCE_NP_COMMUNITY_ERROR_INSUFFICIENT_ARGUMENT	0x8055070c	NULL was specified to <i>idArray</i> , <i>rankArray</i> , <i>totalRecord</i> , or <i>lastSortDate</i> . Or 0 was specified to <i>arrayNum</i>
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	ID specified for <i>reqId</i> does not exist
SCE_NP_COMMUNITY_ERROR_INVALID_ALIGNMENT	0x80550714	Size of structure specified to <i>rankArraySize</i> , <i>commentArraySize</i> , or <i>infoArraySize</i> is invalid
SCE_NP_COMMUNITY_ERROR_TOO_MANY_NPID	0x80550719	Value exceeding 101 was specified to <i>arrayNum</i>
SCE_NP_COMMUNITY_SERVER_ERROR_GAME_RANKING_NOT_FOUND	0x80550812	None of the specified players has scores registered
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_BOARD_MASTER_NOT_FOUND	0x80550817	Scoreboard specified with <i>boardId</i> does not exist
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_BEFORE_SERVICE	0x805508a3	Ranking service of the specified NP Communication ID has not yet started. Unless requested, there is no start date set for the ranking service (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_END_OF_SERVICE	0x805508a4	Ranking service of the specified NP Communication ID has ended (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_MAINTENANCE	0x805508a5	Ranking service of the specified NP Communication ID is under maintenance (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_NO_SUCH_TITLE	0x805508a6	Specified NP Communication ID is not registered on the ranking server

SCE CONFIDENTIAL

Description

This function gets the ranking information of the specified player in the specified scoreboard. When 0 is specified for the player character ID, this function is the same as `sceNpScoreGetRankingByNpId()`. The function is blocked until communication completes and scoreboard information is obtained.

See Also

`sceNpScoreCreateRequest()`, `sceNpScoreAbortRequest()`

000004892117

SCE CONFIDENTIAL

sceNpScoreGetRankingByNpIdPcIdAsync

Get ranking information of player with player character ID specified (asynchronous)

Definition

```
#include <np.h>
int sceNpScoreGetRankingByNpIdPcIdAsync (
    SceInt32 reqId,
    SceNpScoreBoardId boardId,
    const SceNpScoreNpIdPcId *idArray,
    SceSize idArraySize,
    SceNpScorePlayerRankData *rankArray,
    SceSize rankArraySize,
    SceNpScoreComment *commentArray,
    SceSize commentArraySize,
    SceNpScoreGameInfo *infoArray,
    SceSize infoArraySize,
    SceSize arrayNum,
    SceRtcTick *lastSortDate,
    SceNpScoreRankNumber *totalRecord,
    void *option
);
```

Arguments

<i>reqId</i>	NP ScoreRanking request ID (IN)
<i>boardId</i>	Target scoreboard ID (IN)
<i>idArray</i>	Pointer to top array of the target player NP ID structure with player character ID. This area must be 8-byte aligned and continuous (IN)
<i>idArraySize</i>	Memory size of <i>idArray</i> . Specify (<i>arrayNum</i> * sizeof(SceNpScoreNpIdPcId)) (IN)
<i>rankArray</i>	Pointer to top array of structure where ranking information is to be stored. This area must be 8-byte aligned and continuous (OUT)
<i>rankArraySize</i>	Memory size of <i>rankArray</i> . Specify (<i>arrayNum</i> * sizeof(SceNpScorePlayerRankData)) (IN)
<i>commentArray</i>	Pointer to top array of structure where NP ScoreRanking comment is to be stored. This area must be 8-byte aligned and continuous. Comment will not be set when NULL is specified here (OUT)
<i>commentArraySize</i>	Memory size of <i>commentArray</i> . Specify (<i>arrayNum</i> * sizeof(SceNpScoreComment)). This argument will not be referenced when NULL is specified in <i>commentArray</i> (IN)
<i>infoArray</i>	Pointer to top array of structure where data unique to a title is to be stored. This area must be 8-byte aligned and continuous. Data will not be set when NULL is specified here (OUT)
<i>infoArraySize</i>	Memory size of <i>infoArray</i> . Specify (<i>arrayNum</i> * sizeof(SceNpScoreGameInfo)). This argument will not be referenced when NULL is specified in <i>infoArray</i> (IN)
<i>arrayNum</i>	Number of arrays in <i>idArray</i> . Maximum number is 101 (IN)
<i>lastSortDate</i>	Time when server created ranking information (UTC) (OUT)
<i>totalRecord</i>	Total number of players registered on the target scoreboard (OUT)
<i>option</i>	Option reserved for future extension. Always specify NULL

©SCEI

SCE CONFIDENTIAL

Return Values

Returns 0 upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ARGUMENT	0x80550704	Value other than NULL was specified to <i>option</i>
SCE_NP_COMMUNITY_ERROR_INSUFFICIENT_ARGUMENT	0x8055070c	NULL was specified to <i>idArray</i> . Or 0 was specified to <i>arrayNum</i>
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	ID specified for <i>reqId</i> does not exist

Description

This function gets the ranking information of the specified player in the specified scoreboard. When 0 is specified for the player character ID, this function is the same as `sceNpScoreGetRankingByNpIdAsync()`. The actual transaction is executed asynchronously and to receive the result, a function such as `sceNpScoreWaitAsync()` must always be used.

See Also

`sceNpScoreCreateRequest()`, `sceNpScoreAbortRequest()`, `sceNpScoreWaitAsync()`, `sceNpScorePollAsync()`

sceNpScoreGetFriendsRanking

Get top 100 rankings among friends (synchronous)

Definition

```
#include <np.h>
int sceNpScoreGetFriendsRanking (
    SceInt32 reqId,
    SceNpScoreBoardId boardId,
    SceBool includeSelf,
    SceNpScoreRankData *rankArray,
    SceSize rankArraySize,
    SceNpScoreComment *commentArray,
    SceSize commentArraySize,
    SceNpScoreGameInfo *infoArray,
    SceSize infoArraySize,
    SceSize arrayNum,
    SceRtcTick *lastSortDate,
    SceNpScoreRankNumber *totalRecord,
    SceNpScoreGetFriendRankingOptParam *option
);
```

Arguments

<i>reqId</i>	NP ScoreRanking request ID (IN)
<i>boardId</i>	Target scoreboard ID (IN)
<i>includeSelf</i>	Flag to indicate whether to include own ranking information. Specify any number besides 0 to include and 0 to exclude. (IN)
<i>rankArray</i>	Pointer to top array of structure where ranking information is to be stored. This area must be 8-byte aligned and continuous (OUT)
<i>rankArraySize</i>	Memory size of <i>rankArray</i> . Specify (<i>arrayNum</i> * sizeof(SceNpScoreRankData)) (IN)
<i>commentArray</i>	Pointer to top array of structure where NP ScoreRanking comments are to be stored. This area must be 8-byte aligned and continuous. Comments will not be set when NULL is specified here (OUT)
<i>commentArraySize</i>	Memory size of <i>commentArray</i> . Specify (<i>arrayNum</i> * sizeof(SceNpScoreComment)). This argument will not be referenced when NULL is specified in <i>commentArray</i> (IN)
<i>infoArray</i>	Pointer to top array of structure where data unique to a title is to be stored This area must be 8-byte aligned and continuous Data will not be set when NULL is specified here (OUT)
<i>infoArraySize</i>	Memory size of <i>infoArray</i> . Specify (<i>arrayNum</i> * sizeof(SceNpScoreGameInfo)). This argument will not be referenced when NULL is specified in <i>infoArray</i> (IN)
<i>arrayNum</i>	Number of ranking information to obtain. Maximum number is 100 (IN)
<i>lastSortDate</i>	Time when server created ranking information (UTC). This information cannot be obtained if there are 0 numbers of obtainment-target ranking information (return value is 0) (OUT)
<i>totalRecord</i>	Total number of players registered in the target scoreboard. This is not the number of friends registered but the total number of players registered for the game. This information cannot be obtained if there are 0 numbers of obtainment-target ranking information (return value is 0) (OUT)

SCE CONFIDENTIAL

option

Pointer to an extended option structure. Specify NULL if not required (IN/OUT)

Return Values

Returns the number of ranking information obtained (>0) upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ARGUMENT	0x80550704	Value other than NULL was specified to <i>option</i> or value exceeding 100 was specified to <i>arrayNum</i>
SCE_NP_COMMUNITY_ERROR_ABORTED	0x80550707	Transaction was aborted with <i>sceNpScoreAbortRequest()</i> or <i>sceNpScoreDeleteRequest()</i>
SCE_NP_COMMUNITY_ERROR_INVALID_SIGNATURE	0x8055070b	Authentication error. Usually occurs with an error in <i>SceNpCommunicationPassphrase</i>
SCE_NP_COMMUNITY_ERROR_INSUFFICIENT_ARGUMENT	0x8055070c	NULL was specified to <i>rankArray</i> or 0 was specified to <i>arrayNum</i>
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	ID specified for <i>reqId</i> does not exist
SCE_NP_COMMUNITY_ERROR_INVALID_ALIGNMENT	0x80550714	Size of structure specified to <i>rankArraySize</i> , <i>commentArraySize</i> , or <i>infoArraySize</i> is invalid
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_BOARD_MASTER_NOT_FOUND	0x80550817	Scoreboard specified with <i>boardId</i> does not exist
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_BEFORE_SERVICE	0x805508a3	Ranking service of the specified NP Communication ID has not yet started. Unless requested, there is no start date set for the ranking service (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_END_OF_SERVICE	0x805508a4	Ranking service of the specified NP Communication ID has ended (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_RANKING_MAINTENANCE	0x805508a5	Ranking service of the specified NP Communication ID is under maintenance (Corresponding message is in the Message Dialog library)
SCE_NP_COMMUNITY_SERVER_ERROR_NO_SUCH_TITLE	0x805508a6	Specified NP Communication ID is not registered on the ranking server

Description

This function obtains the top ranking information of up to 100 friends for the target scoreboard and stores the information in ranked order to *rankArray*. The number of obtained ranking information will be returned as the return value.

When there is no ranking information to obtain, 0 returns instead of

`SCE_NP_COMMUNITY_SERVER_ERROR_GAME_RANKING_NOT_FOUND`. Note that in this case, values will not be stored in *lastSortDate* and *totalRecord*. Moreover, the value stored to *totalRecord* represents the total number of players registered to the game and not just the number of registered friends.

The function is blocked until communication completes and board information is obtained.

See Also

`sceNpScoreCreateRequest()`, `sceNpScoreAbortRequest()`

sceNpScoreGetFriendsRankingAsync

Get top 100 rankings among friends (asynchronous)

Definition

```
#include <np.h>
int sceNpScoreGetFriendsRankingAsync (
    SceInt32 reqId,
    SceNpScoreBoardId boardId,
    SceBool includeSelf,
    SceNpScoreRankData *rankArray,
    SceSize rankArraySize,
    SceNpScoreComment *commentArray,
    SceSize commentArraySize,
    SceNpScoreGameInfo *infoArray,
    SceSize infoArraySize,
    SceSize arrayNum,
    SceRtcTick *lastSortDate,
    SceNpScoreRankNumber *totalRecord,
    SceNpScoreGetFriendRankingOptParam *option
);
```

Arguments

<i>reqId</i>	NP ScoreRanking request ID (IN)
<i>boardId</i>	Target scoreboard ID (IN)
<i>includeSelf</i>	Flag to indicate whether to include own ranking information. Specify any number besides 0 to include and 0 to exclude (IN)
<i>rankArray</i>	Pointer to top array of structure where ranking information is to be stored. This area must be 8-byte aligned and continuous (OUT)
<i>rankArraySize</i>	Memory size of <i>rankArray</i> . Specify (<i>arrayNum</i> * sizeof(SceNpScoreRankData)) (IN)
<i>commentArray</i>	Pointer to top array of structure where NP ScoreRanking comments are to be stored. This area must be 8-byte aligned and continuous. Comments will not be set when NULL is specified here (OUT)
<i>commentArraySize</i>	Memory size of <i>commentArray</i> . Specify (<i>arrayNum</i> * sizeof(SceNpScoreComment)). This argument will not be referenced when NULL is specified in <i>commentArray</i> (IN)
<i>infoArray</i>	Pointer to top array of structure where data unique to a title is to be stored. This area must be 8-byte aligned and continuous. Data will not be set when NULL is specified here (OUT)
<i>infoArraySize</i>	Memory size of <i>infoArray</i> . Specify (<i>arrayNum</i> * sizeof(SceNpScoreGameInfo)). This argument will not be referenced when NULL is specified in <i>infoArray</i> (IN)
<i>arrayNum</i>	Number of ranking information to obtain. Maximum number is 100 (IN)
<i>lastSortDate</i>	Time when server created ranking information (UTC). This information cannot be obtained if there are 0 numbers of obtainment-target ranking information (return value is 0) (OUT)
<i>totalRecord</i>	Total number of players registered in the target scoreboard. This is not the number of friends registered but the total number of players registered for the game. This information cannot be obtained if there are 0 numbers of obtainment-target ranking information (return value is 0) (OUT)

SCE CONFIDENTIAL

option

Pointer to an extended option structure. Specify NULL if not required (IN/OUT)

Return Values

Returns 0 upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ARGUMENT	0x80550704	Value other than NULL was specified to <i>option</i> or value exceeding 100 was specified to <i>arrayNum</i>
SCE_NP_COMMUNITY_ERROR_INSUFFICIENT_ARGUMENT	0x8055070c	NULL was specified to <i>rankArray</i> or 0 was specified to <i>arrayNum</i> .
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	ID specified for <i>reqId</i> does not exist

Description

This function obtains the top ranking information of up to 100 friends for the target scoreboard and stores the information in ranked order to *rankArray*.

The actual transaction is executed asynchronously and to receive the result, a function such as `sceNpScoreWaitAsync()` must always be used. The number of obtained ranking information will be stored in **result* of `sceNpScoreWaitAsync()/sceNpScorePollAsync()`.

When there is no ranking information to obtain, 0 returns instead of

SCE_NP_COMMUNITY_SERVER_ERROR_GAME_RANKING_NOT_FOUND. Note that in this case, values will not be stored in *lastSortDate* and *totalRecord*. Moreover, the value stored to *totalRecord* represents the total number of players registered to the game and not just the number of registered friends.

See Also

`sceNpScoreCreateRequest()`, `sceNpScoreAbortRequest()`, `sceNpScoreWaitAsync()`, `sceNpScorePollAsync()`

SCE CONFIDENTIAL

sceNpScoreAbortRequest

Abort communication

Definition

```
#include <np.h>
int sceNpScoreAbortRequest(
    SceInt32 reqId
);
```

Arguments

reqId NP ScoreRanking request ID

Return Values

Returns 0 upon normal termination.

Returns a negative value upon 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_COMMUNITY_ERROR_NOT_INITIALIZED	0x80550702	Library is not initialized
SCE_NP_COMMUNITY_ERROR_INVALID_ID	0x8055070e	ID specified for <i>reqId</i> does not exist

Description

This function aborts the applicable communication processing.

Notes

For the following asynchronous functions of the NP ScoreRanking library, you must not destroy the memory specified for storing the request results until either the processing results are obtained using `sceNpScoreWaitAsync()` or `sceNpScorePollAsync()`, or the request is destroyed with `sceNpScoreDeleteRequest()`.

Even when processing is aborted with this function, it is not possible to destroy the specified memory immediately.

- `sceNpScoreGetBoardInfoAsync()`
- `sceNpScoreRecordScoreAsync()`
- `sceNpScoreRecordGameDataAsync()`
- `sceNpScoreGetGameDataAsync()`
- `sceNpScoreGetRankingByNpIdAsync()`
- `sceNpScoreGetRankingByRangeAsync()`

See Also

`sceNpScoreCreateRequest()`, `sceNpScoreDeleteRequest()`, `sceNpScoreWaitAsync()`, `sceNpScorePollAsync()`

Constants

000004892117

SCE CONFIDENTIAL

SCE_NP_SCORE_COMMENT_MAXLEN

Maximum length of ranking comment

Definition

```
#include <np.h>

#define SCE_NP_SCORE_COMMENT_MAXLEN 63
```

Description

This constant represents the maximum length of a ranking comment. Because ranking comments are set in UTF-8, length is represented in bytes rather than the length of character strings.

See Also

SceNpScoreComment

SCE CONFIDENTIAL

SCE_NP_SCORE_CENSOR_COMMENT_MAXLEN

Maximum length for a comment to be checked

Definition

```
#include <np.h>

#define SCE_NP_SCORE_CENSOR_COMMENT_MAXLEN 255
```

Description

This constant represents the maximum length of a comment that can be checked by `sceNpScoreCensorComment()` or `sceNpScoreCensorCommentAsync()`. Because comments are set in UTF-8, length is represented in bytes rather than the length of character strings.

See Also

`sceNpScoreCensorComment()`, `sceNpScoreCensorCommentAsync()`

SCE CONFIDENTIAL

SCE_NP_SCORE_SANITIZE_COMMENT_MAXLEN

Maximum length for a comment to be corrected

Definition

```
#include <np.h>

#define SCE_NP_SCORE_SANITIZE_COMMENT_MAXLEN 255
```

Description

This constant represents the maximum length of a comment that can be corrected by `sceNpScoreSanitizeComment()` or `sceNpScoreSanitizeCommentAsync()`. Because comments are set in UTF-8, length is represented in bytes rather than the length of character strings.

See Also

`sceNpScoreSanitizeComment()`, `sceNpScoreSanitizeCommentAsync()`

SCE CONFIDENTIAL

SCE_NP_SCORE_GAMEINFO_MAXSIZE

Maximum size of SceNpScoreGameInfo

Definition

```
#include <np.h>

#define SCE_NP_SCORE_GAMEINFO_MAXSIZE 189
```

Description

This constant represents the maximum size of the data stored in `SceNpScoreGameInfo`. Actual valid data size is stored in `infoSize` member.

See Also

`SceNpScoreGameInfo`

SCE CONFIDENTIAL

SCE_NP_SCORE_MAX_CTX_NUM

Maximum number of contexts

Definition

```
#include <np.h>

#define SCE_NP_SCORE_MAX_CTX_NUM    (32)
```

Description

This constant indicates the maximum number of contexts that can exist at one time in the NP ScoreRanking library.

This number is both the maximum for title contexts and the maximum for requests.

See Also

sceNpScoreCreateTitleCtx(), sceNpScoreCreateRequest()

SCE CONFIDENTIAL

SCE_NP_SCORE_MAX_RANGE_NUM_PER_REQUEST

Maximum range of scores obtained at one time (when the rank is specified)

Definition

```
#include <np.h>

#define SCE_NP_SCORE_MAX_RANGE_NUM_PER_REQUEST (100)
```

Description

This constant indicates the maximum range of scores that can be obtained at one time in a transaction where the ranking is specified to obtain ranking information.

See Also

`sceNpScoreGetRankingByRange()`, `sceNpScoreGetRankingByRangeAsync()`

SCE CONFIDENTIAL

SCE_NP_SCORE_MAX_NPID_NUM_PER_REQUEST

Maximum number of players that can be specified at one time to obtain ranking information

Definition

```
#include <np.h>

#define SCE_NP_SCORE_MAX_NPID_NUM_PER_REQUEST    (101)
```

Description

This constant indicates the maximum number of players that can be specified at one time in a transaction where the player is specified to obtain ranking information.

See Also

```
sceNpScoreGetRankingByNpId(), sceNpScoreGetRankingByNpIdAsync(),
sceNpScoreGetRankingByNpIdPcId(), sceNpScoreGetRankingByNpIdPcIdAsync()
```


SCE CONFIDENTIAL

SCE_NP_SCORE_MAX_SELECTED_FRIENDS_NUM

Maximum number of users that can be obtained with communication processing that targets friends

Definition

```
#include <np.h>

#define SCE_NP_SCORE_MAX_SELECTED_FRIENDS_NUM_ (100)
```

Description

This constant indicates the maximum number of target users to obtain that can be specified in communication processing that targets friends.

See Also

`sceNpScoreGetFriendsRanking()`, `sceNpScoreGetFriendsRankingAsync()`