

NP Party Library Reference

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

Table of Contents

Datatypes.....	4
SceNpPartyEventHandlers	5
SceNpPartyGameSessionEvent	6
SceNpPartyGameSessionMemberUpdateType	7
SceNpPartyGameSessionUpdateReady	8
SceNpPartyId	9
SceNpPartyInitParam.....	10
SceNpPartyMemberFlags	11
SceNpPartyMemberGameSessionUpdateInfo	12
SceNpPartyMemberInfo.....	13
SceNpPartyMemberList	14
SceNpPartyMemberVoiceInfo	15
SceNpPartyMemberVoiceState.....	16
SceNpPartyRoomEventType	17
SceNpPartyRoomLeftReason	18
SceNpPartyRoomMemberId	19
SceNpPartyState	20
Functions.....	21
Library Initialization and Termination.....	22
sceNpPartyInit.....	22
sceNpPartyInitParamInit	23
sceNpPartyTerm.....	24
Event Handling	25
sceNpPartyCheckCallback	25
sceNpPartyEventHandlersInit.....	26
sceNpPartyRegisterHandler	27
Request	28
sceNpPartyGetGameSessionReadyState	28
sceNpPartyGetId.....	29
sceNpPartyGetMemberInfo	30
sceNpPartyGetMembers.....	31
sceNpPartyGetMemberSessionInfo.....	32
sceNpPartyGetMemberVoiceInfo.....	33
sceNpPartyGetState.....	34
Callback Functions.....	35
SceNpPartyGameSessionEventHandler	36
SceNpPartyRoomEventHandler	37
SceNpPartyVoiceEventHandler	38
Error Codes	39
Define Summary	40
Other Constants.....	41
Party Room Event Types.....	42
Define Summary	42

SCE CONFIDENTIAL

Party Member Voice States	43
Define Summary	43
Party States	44
Define Summary	44
Party Room Limits	45
Define Summary	45
Party Member Flags	46
Define Summary	46
Party Room Reason for Leaving Flags	47
Define Summary	47
Party Game-Session Events	48
Define Summary	48
Party Game-Session Update Types	49
Define Summary	49

Document serial number: 000004892117

Datatypes

000004892117

SCE CONFIDENTIAL

SceNpPartyEventHandlers

Structure for registering event handlers.

Definition

```
#include <np/np_party.h>
typedef struct SceNpPartyEventHandlers {
    SceUInt32 sdkVersion;
    SceNpPartyRoomEventHandler roomEventHandler;
    SceNpPartyVoiceEventHandler voiceEventHandler;
    SceNpPartyGameSessionEventHandler gameSessionEventHandler;
} SceNpPartyEventHandlers;
```

Members

<i>sdkVersion</i>	SDK version.
<i>roomEventHandler</i>	Event handler for subscribing to party room events.
<i>voiceEventHandler</i>	Event handler for subscribing to party voice notification events.
<i>gameSessionEventHandler</i>	Event handler for subscribing to game-session change events for party members.

Description

This structure is used for registering event handlers. It is passed as an argument of [sceNpPartyRegisterHandler\(\)](#). If you do not want to use event handlers, set the applicable member to NULL.

Notes

This structure must be initialized with [sceNpPartyEventHandlersInit\(\)](#).

See Also

[SceNpPartyRoomEventHandler](#), [SceNpPartyVoiceEventHandler](#),
[sceNpPartyRegisterHandler\(\)](#), [sceNpPartyEventHandlersInit\(\)](#)

SCE CONFIDENTIAL

SceNpPartyGameSessionEvent

Type indicating party game-session events.

Definition

```
#include <np/np_party.h>
typedef SceUInt8 SceNpPartyGameSessionEvent;
```

Description

This type represents the kind of event received from the party game-session event handler. The party game-session events are represented by the following constants:

Constant	Description
SCE_NP_PARTY_GAME_SESSION_UPDATE_EVENT_READY	If this event is received and the event handler returns SCE_OK, the application will receive SCE_NP_PARTY_GAME_SESSION_MEMBER_UPDATE_EVENT events. The data field should be cast to an SceNpPartyGameSessionUpdateReady structure.
SCE_NP_PARTY_GAME_SESSION_MEMBER_UPDATE_EVENT	The event handler has a game-session update event for a party member. The application should cast the data field into a struct object of type SceNpPartyMemberGameSessionUpdateInfo .

SCE CONFIDENTIAL

SceNpPartyGameSessionMemberUpdateType

Type indicating party game-session update for a party member.

Definition

```
#include <np/np_party.h>
typedef SceUInt8 SceNpPartyGameSessionMemberUpdateType;
```

Description

This type represents whether the party member is in a joinable game with the current application. To be joinable, the game must be compatible with the application that is locally using the NP Party library.

The game-session member update types are represented by the following constants:

Constant	Description
SCE_NP_PARTY_GAME_SESSION_UPDATE_TYPE_NOT_JOINABLE	The party member is not in a joinable session with the application using the NP Party library.
SCE_NP_PARTY_GAME_SESSION_UPDATE_TYPE_JOINABLE	The party member is in a joinable session with the application using the NP Party library.

SCE CONFIDENTIAL

SceNpPartyGameSessionUpdateReady

Structure containing the party game session update-ready state.

Definition

```
#include <np/np_party.h>
typedef struct SceNpPartyGameSessionUpdateReady {
    SceUInt32 result;
} SceNpPartyGameSessionUpdateReady;
```

Members

<i>result</i>	If the result is <code>SCE_OK</code> , the application will receive game-session updates for party members.
---------------	---

Description

This structure contains the party game session update-ready state. When this event is received, the status should be checked by the application. If the result is `SCE_OK`, the application can receive party game-session updates. Each time the library is initialized, the application must wait for this event to be received by the callback. If the result is not `SCE_OK` (which is rare and should not occur), the application must terminate and re-initialize the party in order to attempt to use party game-session updates.

See Also

[SceNpPartyGameSessionEventHandler](#), [SceNpPartyGameSessionEvent](#),
`SCE_NP_PARTY_MEMBER_GAME_SESSION_EVENT_UPDATE`

SCE CONFIDENTIAL

SceNpPartyId

Structure for receiving the party ID.

Definition

```
#include <np/np_party.h>
typedef struct SceNpPartyId {
    SceUChar8 id[SCE_NP_PARTY_ID_LENGTH];
} SceNpPartyId;
```

Members

id Unique identifier for the party.

Description

This structure contains a unique identifier for the party. This party room ID is guaranteed to be unique across parties at all times.

See Also

[sceNpPartyGetId\(\)](#)

SCE CONFIDENTIAL

SceNpPartyInitParam

Structure containing NP Party initialization parameters.

Definition

```
#include <np/np_party.h>
typedef struct SceNpPartyInitParam {
    SceUInt32 sdkVersion;
    SceNpCommunicationId commId;
} SceNpPartyInitParam;
```

Members

<i>sdkVersion</i>	SDK version.
<i>commId</i>	The application's NP Communication ID. Deprecated.

Description

This structure represents the initialization parameters for the NP Party library.

Notes

This structure must be initialized with [sceNpPartyInitParamInit\(\)](#).

See Also

[sceNpPartyInitParamInit\(\)](#)

SCE CONFIDENTIAL

SceNpPartyMemberFlags

Type indicating party member flags.

Definition

```
#include <np/np_party.h>
typedef SceUInt8 SceNpPartyMemberFlags;
```

Description

This type represents the flags of the party member. The member flags are represented by the following constants:

Constant	Description
SCE_NP_PARTY_MEMBER_FLAG_IS_ME	The member is the current user.
SCE_NP_PARTY_MEMBER_FLAG_IS_PARTY_LEADER	The member is the party leader.

SCE CONFIDENTIAL

SceNpPartyMemberGameSessionUpdateInfo

Structure containing a party member's game-session update.

Definition

```
#include <np/np_party.h>
typedef struct SceNpPartyMemberGameSessionUpdateInfo {
    SceNpPartyGameSessionMemberUpdateType type;
    SceUInt8 reserved;
    SceNpPartyRoomMemberId memberId;
    SceUInt8 reserved2[2];
    SceNpSessionId sessionId;
} SceNpPartyMemberGameSessionUpdateInfo;
```

Members

<i>type</i>	The game-session update information for the party member specified in <i>memberId</i> .
<i>reserved</i>	Reserved field; do not use.
<i>memberId</i>	The member ID of the party member related to this game-session update.
<i>reserved2</i>	Reserved field; do not use.
<i>sessionId</i>	Session ID of the game being played by the party member specified in <i>memberId</i> . Only valid when <i>type</i> is SCE NP PARTY GAME SESSION UPDATE TYPE JOINABLE .

Description

This structure contains general information pertaining to a party member's joinable game session member.

See Also

[SCE NP PARTY GAME SESSION MEMBER UPDATE EVENT](#)

SCE CONFIDENTIAL

SceNpPartyMemberInfo

Structure containing general party member information.

Definition

```
#include <np/np_party.h>
typedef struct SceNpPartyMemberInfo {
    SceNpId npId;
    SceNpPartyRoomMemberId memberId;
    SceNpPartyMemberFlags memberFlags;
    SceUChar8 reserved;
    SceNpBasicGamePresence gamePresence;
    SceNpCommunicationId commId;
} SceNpPartyMemberInfo;
```

Members

<i>npId</i>	NP ID of the party member.
<i>memberId</i>	Member ID of the party member.
<i>memberFlags</i>	Member flags of the party member.
<i>reserved</i>	Reserved for internal use of NP Party library.
<i>gamePresence</i>	Game presence of the party member. Deprecated; use sceNpPartyGetMemberSessionInfo() instead of this struct member.
<i>commId</i>	NP Communication ID of the party member's game. Deprecated; use sceNpPartyGetMemberSessionInfo() instead of this struct member.

Description

This structure contains general information pertaining to a party member.

See Also

[SceNpPartyMemberFlags](#), [SceNpPartyRoomMemberId](#), [SceNpBasicGamePresence](#), [SceNpBasicInGamePresenceType](#), [SceNpCommunicationId](#) in the *NP Basic Library Reference*.

SCE CONFIDENTIAL

SceNpPartyMemberList

Structure containing the party member list.

Definition

```
#include <np/np_party.h>
typedef struct SceNpPartyMemberList {
    SceNpId members[SCE_NP_PARTY_MEMBER_NUM_MAX];
    SceUInt8 memberNum;
    bool privateParty;
    SceUChar8 padding[2];
} SceNpPartyMemberList;
```

Members

<i>members</i>	Array of NP IDs for party members.
<i>memberNum</i>	Current count of members in the party.
<i>privateParty</i>	Flag that when set to true indicates that only invited party members are allowed in the party.
<i>padding</i>	Padding to keep the structure 4-byte aligned.

Description

This structure lists the NP IDs of all party members. It can include as many as [SCE_NP_PARTY_MEMBER_NUM_MAX](#) members, where [SCE_NP_PARTY_MEMBER_NUM_MAX](#) is defined as 8.

SCE CONFIDENTIAL

SceNpPartyMemberVoiceInfo

Structure containing party member voice information.

Definition

```
#include <np/np_party.h>
typedef struct SceNpPartyMemberVoiceInfo {
    SceNpPartyRoomMemberId memberId;
    SceNpPartyMemberVoiceState memberVoiceState;
    SceUChar8 padding;
} SceNpPartyMemberVoiceInfo;
```

Members

<i>memberId</i>	Member ID of the party member.
<i>memberVoiceState</i>	Voice state of the party member. One of the states defined by SceNpPartyMemberVoiceState .
<i>padding</i>	Padding to keep the structure 4-byte aligned.

Description

This structure contains voice information pertaining to a party member.

See Also

[SceNpPartyMemberVoiceState](#), [SceNpPartyRoomMemberId](#)

SCE CONFIDENTIAL

SceNpPartyMemberVoiceState

Type indicating the party member voice state.

Definition

```
#include <np/np_party.h>
typedef SceUChar8 SceNpPartyMemberVoiceState;
```

Description

This type represents the party member voice state. The party member voice states are represented by the following constants:

Constant	Description
SCE_NP_PARTY_MEMBER_VOICE_STATE_UNKNOWN	A member's voice peer-to-peer connection is unknown and being established.
SCE_NP_PARTY_MEMBER_VOICE_STATE_CONNECTED	A member's voice peer-to-peer connection has been established. This event represents the state of a party member that is not currently talking.
SCE_NP_PARTY_MEMBER_VOICE_STATE_DISCONNECTED	A member's voice peer-to-peer connection has been disconnected.
SCE_NP_PARTY_MEMBER_VOICE_STATE_MUTED	A remote party member has been muted in the party application.
SCE_NP_PARTY_MEMBER_VOICE_STATE_TALKING	A party member is talking. There is no corresponding event to indicate when the party member has finished talking; so the game should use a timer to expire a party member from a talking state. It is recommended that games indicate that a party member is in a talking state for up to one second after an SCE_NP_PARTY_MEMBER_VOICE_STATE_TALKING voice event occurs.

SCE CONFIDENTIAL

SceNpPartyRoomEventType

Type indicating a party room event.

Definition

```
#include <np/np_party.h>
typedef SceUInt16 SceNpPartyRoomEventType;
```

Description

This type represents the type of party room event that occurred. The party room events are represented by the following constants:

Constant	Description
SCE_NP_PARTY_ROOM_EVENT_JOINED	The current user joined a party.
SCE_NP_PARTY_ROOM_EVENT_MEMBER_JOINED	A member joined the party that the current user is in.
SCE_NP_PARTY_ROOM_EVENT_MEMBER_LEFT	A member left the party that the current user is in.
SCE_NP_PARTY_ROOM_EVENT_LEFT	The current user left the party.
SCE_NP_PARTY_ROOM_EVENT_MEMBER_PRESENCE_UPDATE	A member in the party that the current user is in has updated their presence.

SCE CONFIDENTIAL

SceNpPartyRoomLeftReason

Type indicating the reason a member left the party room.

Definition

```
#include <np/np_party.h>
typedef SceUInt32 SceNpPartyRoomLeftReason;
```

Description

This type represents the possible reasons a member leaves the party room. The reasons are represented by the following constants:

Constant	Description
SCE_NP_PARTY_ROOM_LEFT_REASON_USER_ACTION	Party member left due to the member's action, such as quitting the party.
SCE_NP_PARTY_ROOM_LEFT_REASON_SERVER_ERROR	Party member left due to a server error.
SCE_NP_PARTY_ROOM_LEFT_REASON_KICKED_OUT	Party member left due to being kicked out of the party.

SCE CONFIDENTIAL

SceNpPartyRoomMemberId

Type indicating a party member ID.

Definition

```
#include <np/np_party.h>
typedef SceUInt16 SceNpPartyRoomMemberId;
```

Description

This type represents the unique ID of the party member.

000004892117

SCE CONFIDENTIAL

SceNpPartyState

Type indicating the party room state.

Definition

```
#include <np/np_party.h>
typedef SceUInt16 SceNpPartyState;
```

Description

This type represents the party room state. The party room states are represented by the following constants:

Constant	Description
SCE_NP_PARTY_STATE_IN_PARTY	The current user is in the party.
SCE_NP_PARTY_STATE_NOT_IN_PARTY	The current user is not in the party.
SCE_NP_PARTY_STATE_IN_PRIVATE_PARTY	The current user is in a private party.

Functions

000004892117

Library Initialization and Termination

sceNpPartyInit

Initializes the NP Party library.

Definition

```
#include <np/np_party.h>
int sceNpPartyInit(
    const SceNpPartyInitParam *param
);
```

Calling Conditions

Not multithread safe.

Arguments

param NP Party initialization parameters structure.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_NP_PARTY_ERROR_UNKNOWN	Unknown error.
SCE_NP_PARTY_ERROR_ALREADY_INITIALIZED	The NP Party library is already initialized.
SCE_NP_PARTY_ERROR_INVALID_ARGUMENT	Reserved parameter is not zeroed.

Description

This function initializes the NP Party library. Call [sceNpPartyInitParamInit\(\)](#) first to setup an [SceNpPartyInitParam](#) structure; update the [SceNpPartyInitParam](#) structure as appropriate for the application; then call [sceNpPartyInit\(\)](#) with the updated data.

Notes

Because this function communicates with system processes, it can block other processes. The length of time it blocks depends on the system process load. Make sure to call this function from a thread that can tolerate the effects of the system process load; for example, do not call this function on the rendering thread.

See Also

[SceNpPartyInitParam](#), [sceNpPartyTerm\(\)](#)

sceNpPartyInitParamInit

Initializes the NP Party initialization parameters structure.

Definition

```
#include <np/np_party.h>
static inline void sceNpPartyInitParamInit(
    SceNpPartyInitParam *param
);
```

Arguments

param NP Party initialization parameters structure to be initialized.

Return Values

None

Description

This inline function initializes the NP Party initialization parameters structure. Call this function first to setup an [SceNpPartyInitParam](#) structure; update the [SceNpPartyInitParam](#) structure as appropriate for the application; then call [sceNpPartyInit\(\)](#) with the updated data.

See Also

[SceNpPartyInitParam](#)

SCE CONFIDENTIAL

sceNpPartyTerm

Terminates the NP Party library.

Definition

```
#include <np/np_party.h>
int sceNpPartyTerm(void);
```

Calling Conditions

Not multithread safe.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_NP_PARTY_ERROR_UNKNOWN	Unknown error.
SCE_NP_PARTY_ERROR_NOT_INITIALIZED	The NP Party library is not initialized.

Description

This function terminates the NP Party library.

Notes

Because this function communicates with system processes, it can block other processes. The length of time it blocks depends on the system process load. Make sure to call this function from a thread that can tolerate the effects of the system process load; for example, do not call this function on the rendering thread.

See Also

[sceNpPartyInit\(\)](#)

Event Handling

sceNpPartyCheckCallback

Calls event handlers when an event occurs.

Definition

```
#include <np/np_party.h>
int sceNpPartyCheckCallback(void);
```

Calling Conditions

Not multithread safe.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_NP_PARTY_ERROR_NOT_INITIALIZED	The NP Party library is not initialized.

Description

This function checks for NP Party event occurrences. Call this function regularly after registering an event handler using [sceNpPartyRegisterHandler\(\)](#). If an event has occurred, the NP Party library calls that event's event handler.

Notes

The event handler is executed in the context of the thread that calls this function.

See Also

[sceNpPartyRegisterHandler\(\)](#)

SCE CONFIDENTIAL

sceNpPartyEventHandlersInit

Initializes the event handler structure.

Definition

```
#include <np/np_party.h>
static inline void sceNpPartyEventHandlersInit(
    SceNpPartyEventHandlers *handlers
);
```

Arguments

handlers The event handler structure.

Return Values

None

Description

This inline function is used to initialize the event handler structure. Call this function before using [sceNpPartyRegisterHandler\(\)](#).

See Also

[SceNpPartyEventHandlers](#), [sceNpPartyRegisterHandler\(\)](#)

SCE CONFIDENTIAL

sceNpPartyRegisterHandler

Registers event handlers.

Definition

```
#include <np/np_party.h>
int sceNpPartyRegisterHandler(
    const SceNpPartyEventHandlers *handlers,
    void *userdata
);
```

Calling Conditions

Not multithread safe.

Arguments

handlers Structure for the event handler registration.
userdata Any user data to pass to the event handler.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_NP_PARTY_ERROR_NOT_INITIALIZED	The NP Party library is not initialized.

Description

Call this function to register event handlers.

Notes

Because this function communicates with system processes, it can block other processes. The length of time it blocks depends on the system process load. Make sure to call this function from a thread that can tolerate the effects of the system process load; for example, do not call this function on the rendering thread.

See Also

[SceNpPartyEventHandlers](#)

Request

sceNpPartyGetGameSessionReadyState

Retrieves the structure containing the party game session update-ready state.

Definition

```
#include <np/np_party.h>
int sceNpPartyGetGameSessionReadyState(
    SceNpPartyGameSessionUpdateReady *readyState
);
```

Calling Conditions

Not multithread safe.

Arguments

readyState Structure containing the party game session update-ready state.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_NP_PARTY_ERROR_NOT_INITIALIZED	The NP Party library is not initialized.
SCE_NP_PARTY_ERROR_INVALID_ARGUMENT	One or more NULL parameters.
SCE_NP_PARTY_ERROR_GAME_SESSION_NOT_ENABLED	The SceNpPartyGameSessionEventHandler is not registered.
SCE_NP_PARTY_ERROR_OPERATION_IN_PROGRESS	The party game session update-ready system is in progress and will finish when the SceNpPartyGameSessionEventHandler is called.

Description

This function retrieves the structure containing the party game session update-ready state.

Notes

Because this function communicates with system processes, it can block other processes. The length of time it blocks depends on the system process load. Make sure to call this function from a thread that can tolerate the effects of the system process load; for example, do not call this function on the rendering thread.

See Also

[sceNpPartyRegisterHandler\(\)](#), [SceNpPartyGameSessionEventHandler](#), [SceNpPartyGameSessionUpdateReady](#)

SCE CONFIDENTIAL

sceNpPartyGetId

Gets the party ID.

Definition

```
#include <np/np_party.h>
int sceNpPartyGetId(
    SceNpPartyId *const id
);
```

Calling Conditions

Not multithread safe.

Arguments

id [SceNpPartyId](#) of the party.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_NP_PARTY_ERROR_UNKNOWN	Unknown error.
SCE_NP_PARTY_ERROR_NOT_INITIALIZED	The NP Party library is not initialized.
SCE_NP_PARTY_ERROR_INVALID_ARGUMENT	One or more NULL parameters.
SCE_NP_PARTY_ERROR_NOT_IN_PARTY	No Local users are in a party.

Description

This function retrieves the party room ID. Because all local users must be in the same party, if any of the local users are in a party, a party room ID will be returned.

See Also

[SceNpPartyId](#)

SCE CONFIDENTIAL

sceNpPartyGetMemberInfo

Retrieves party member details.

Definition

```
#include <np/np_party.h>
int sceNpPartyGetMemberInfo (
    const SceNpId *npId,
    SceNpPartyMemberInfo *memberInfo
);
```

Calling Conditions

Not multithread safe.

Arguments

<i>npId</i>	NP ID of party member.
<i>memberInfo</i>	Party member information structure.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_NP_PARTY_ERROR_UNKNOWN	Unknown error.
SCE_NP_PARTY_ERROR_NOT_INITIALIZED	The NP Party library is not initialized.
SCE_NP_PARTY_ERROR_INVALID_ARGUMENT	One or more NULL parameters.
SCE_NP_PARTY_ERROR_NOT_IN_A_PARTY	The member is not in a party.
SCE_NP_PARTY_ERROR_PARTY_MEMBER_NOT_FOUND	The party member with the specified NP ID was not found.

Description

This obtains the party member information from a valid NP ID.

Notes

Because this function communicates with system processes, it can block other processes. The length of time it blocks depends on the system process load. Make sure to call this function from a thread that can tolerate the effects of the system process load; for example, do not call this function on the rendering thread.

See Also

[SceNpPartyMemberInfo](#), [sceNpPartyGetMembers\(\)](#)

SCE CONFIDENTIAL

sceNpPartyGetMembers

Retrieves the list of current party members.

Definition

```
#include <np/np_party.h>
int sceNpPartyGetMembers(
    SceNpPartyMemberList *memberList
);
```

Calling Conditions

Not multithread safe.

Arguments

memberList NP Party member list structure.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_NP_PARTY_ERROR_UNKNOWN	Unknown error.
SCE_NP_PARTY_ERROR_NOT_INITIALIZED	The NP Party library is not initialized.
SCE_NP_PARTY_ERROR_INVALID_ARGUMENT	One or more NULL parameters.
SCE_NP_PARTY_ERROR_NOT_IN_A_PARTY	The member is not in a party.

Description

This function obtains the list of current party members.

Notes

Because this function communicates with system processes, it can block other processes. The length of time it blocks depends on the system process load. Make sure to call this function from a thread that can tolerate the effects of the system process load; for example, do not call this function on the rendering thread.

See Also

[SceNpPartyMemberList](#), [sceNpPartyInit\(\)](#)

SCE CONFIDENTIAL

sceNpPartyGetMemberSessionInfo

Retrieves party member game-session details.

Definition

```
#include <np/np_party.h>
int sceNpPartyGetMemberSessionInfo (
    const SceNpPartyRoomMemberId memberId,
    SceNpPartyMemberGameSessionUpdateInfo* memberGameSessionInfo
);
```

Calling Conditions

Not multithread safe.

Arguments

memberId Party member ID ([SceNpPartyRoomMemberId](#)) used to retrieve game-session details.

memberGameSessionInfo Party member game-session information.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_NP_PARTY_ERROR_UNKNOWN	Unknown error.
SCE_NP_PARTY_ERROR_NOT_INITIALIZED	The NP Party library is not initialized.
SCE_NP_PARTY_ERROR_GAME_SESSION_NOT_ENABLED	SceNpPartyGameSessionEventHandler is not registered.
SCE_NP_PARTY_ERROR_INVALID_ARGUMENT	One or more NULL parameters.
SCE_NP_PARTY_ERROR_NOT_IN_PARTY	No local users are in a party.
SCE_NP_PARTY_ERROR_PARTY_MEMBER_NOT_FOUND	The party member with the specified SceNpPartyRoomMemberId was not found.

Description

This function obtains a party member game session from a valid [SceNpPartyRoomMemberId](#). This function will only return the session ID details if the return value is SCE_OK.

Notes

Because this function communicates with system processes, it can block other processes. The length of time during which it blocks other processes depends on the system process load. Make sure to call this function from a thread that can tolerate the effects of the system process load; for example, do not call this function on the rendering thread.

See Also

[SceNpPartyMemberGameSessionUpdateInfo](#), [SceNpPartyRoomMemberId](#), [SceNpPartyGameSessionEventHandler](#)

SCE CONFIDENTIAL

sceNpPartyGetMemberVoiceInfo

Retrieves party member voice details.

Definition

```
#include <np/np_party.h>
int sceNpPartyGetMemberVoiceInfo (
    const SceNpPartyRoomMemberId memberId,
    SceNpPartyMemberVoiceState *memberVoiceState
);
```

Calling Conditions

Not multithread safe.

Arguments

memberId [SceNpPartyRoomMemberId](#) of party member.
memberVoiceState [SceNpPartyMemberVoiceState](#) data.

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_NP_PARTY_ERROR_UNKNOWN	Unknown error.
SCE_NP_PARTY_ERROR_NOT_INITIALIZED	The NP Party library is not initialized.
SCE_NP_PARTY_ERROR_VOICE_NOT_ENABLED	Either SceNpPartyVoiceEventHandler is not registered or there was an SCE_NP_PARTY_GAME_SESSION_UPDATE_EVENT_READY error.
SCE_NP_PARTY_ERROR_INVALID_ARGUMENT	One or more NULL parameters.
SCE_NP_PARTY_ERROR_NOT_IN_A_PARTY	The member is not in a party.
SCE_NP_PARTY_ERROR_PARTY_MEMBER_NOT_FOUND	The party member with the specified SceNpPartyRoomMemberId was not found.

Description

This function obtains party member voice information from a valid [SceNpPartyRoomMemberId](#).

Notes

Because this function communicates with system processes, it can block other processes. The length of time it blocks depends on the system process load. Make sure to call this function from a thread that can tolerate the effects of the system process load; for example, do not call this function on the rendering thread.

See Also

[SceNpPartyMemberVoiceState](#), [SceNpPartyRoomMemberId](#),
[SceNpPartyVoiceEventHandler](#)

SCE CONFIDENTIAL

sceNpPartyGetState

Retrieves the party state of the current user.

Definition

```
#include <np/np_party.h>
int sceNpPartyGetState(
    SceNpPartyState *state
);
```

Calling Conditions

Not multithread safe.

Arguments

state NP Party state structure

Return Values

Value	Description
SCE_OK	Successful completion.
SCE_NP_PARTY_ERROR_UNKNOWN	Unknown error.
SCE_NP_PARTY_ERROR_NOT_INITIALIZED	The NP Party library is not initialized.
SCE_NP_PARTY_ERROR_INVALID_ARGUMENT	One or more NULL parameters.

Description

This function retrieves the party state of the user.

Notes

Because this function communicates with system processes, it can block other processes. The length of time it blocks depends on the system process load. Make sure to call this function from a thread that can tolerate the effects of the system process load; for example, do not call this function on the rendering thread.

See Also

[sceNpPartyInit\(\)](#), [SceNpPartyState](#)

Callback Functions

000004892117

SCE CONFIDENTIAL

SceNpPartyGameSessionEventHandler

Event handler for receiving party member game-session updates.

Definition

```
#include <np/np_party.h>
typedef void (*SceNpPartyGameSessionEventHandler) (
    SceNpPartyGameSessionEvent event,
    const void *data,
    void *userdata
);
```

Arguments

<i>event</i>	The type of event being returned. If <i>event</i> is <code>SCE_NP_PARTY_GAME_SESSION_UPDATE_EVENT_READY</code> , the <i>data</i> field should be cast to SceNpPartyGameSessionUpdateReady . If <i>event</i> is <code>SCE_NP_PARTY_GAME_SESSION_MEMBER_UPDATE_EVENT</code> , the <i>data</i> field should be cast to SceNpPartyMemberGameSessionUpdateInfo .
<i>data</i>	The data field should be cast to the appropriate type as specified by <i>event</i> .
<i>userdata</i>	User data that is set when the event handler is registered.

Return Values

None

Description

This event handler handles game-session update events for remote party members.

When the event handler is registered using [sceNpPartyRegisterHandler\(\)](#), and the `SCE_NP_PARTY_GAME_SESSION_UPDATE_EVENT_READY` is received by the application, NpParty will generate party-member game-session update events for notifying the application of a party member's game-session information. When the application calls [sceNpPartyCheckCallback\(\)](#) and an event has occurred, the event handler is called.

Notes

Ensure the immediate return of the event handler; for example, avoid processing inside the handler that might delay its return. The data must be copied from this structure during the callback. Do not reference *data* after the callback returns.

See Also

[SceNpPartyMemberGameSessionUpdateInfo](#), [SceNpPartyGameSessionUpdateReady](#), [sceNpPartyCheckCallback\(\)](#), [sceNpPartyRegisterHandler\(\)](#), [SceNpPartyGameSessionEvent](#)

SCE CONFIDENTIAL

SceNpPartyRoomEventHandler

Event handler for receiving party room events.

Definition

```
#include <np/np_party.h>
typedef void (*SceNpPartyRoomEventHandler) (
    SceNpPartyRoomEventType eventType,
    const void *data,
    void *userdata
);
```

Arguments

<i>eventType</i>	Generated event type.
<i>data</i>	Event data.
<i>userdata</i>	User data set upon registering the event handler.

Return Values

None

Description

This event handler handles events related to the party room.

When the event handler is registered using [sceNpPartyRegisterHandler\(\)](#), NP Party generates events for notifying the application of various party room events as described by [SceNpPartyRoomEventType](#). When the application calls [sceNpPartyCheckCallback\(\)](#) and an event has occurred, the event handler is called.

The application must cast the *data* pointer to the appropriate structure to retrieve the associated data of the event.

SceNpPartyRoomEventType	Datatype
SCE_NP_PARTY_ROOM_EVENT_JOINED	SceNpPartyMemberList*
SCE_NP_PARTY_ROOM_EVENT_MEMBER_JOINED	SceNpPartyMemberInfo*
SCE_NP_PARTY_ROOM_EVENT_MEMBER_LEFT	SceNpPartyMemberInfo*
SCE_NP_PARTY_ROOM_EVENT_LEFT	SceNpPartyRoomLeftReason*

Notes

Ensure the immediate return of the event handler: for example, avoid processing inside the handler that might delay its return.

The memory area pointed to by *data* will be released after the event handler returns; if needed, the *data* value must be copied onto memory within the handler by the application.

See Also

[SceNpPartyRoomEventType](#), [sceNpPartyCheckCallback\(\)](#),
[sceNpPartyRegisterHandler\(\)](#)

SCE CONFIDENTIAL

SceNpPartyVoiceEventHandler

Event handler for receiving party voice notification events.

Definition

```
#include <np/np_party.h>
typedef void (*SceNpPartyVoiceEventHandler) (
    const SceNpPartyMemberVoiceInfo *memberVoiceInfo,
    void *userdata
);
```

Arguments

<i>memberVoiceInfo</i>	Generated event type.
<i>userdata</i>	User data set upon registering the event handler.

Return Values

None

Description

This event handler handles voice notification events related to the party room.

When the event handler is registered using [sceNpPartyRegisterHandler\(\)](#), NP Party generates voice notification events for notifying the application of when various members of the party are speaking. When the application calls [sceNpPartyCheckCallback\(\)](#) and an event has occurred, the event handler is called.

Notes

Ensure the immediate return of the event handler; for example, avoid processing inside the handler that might delay its return.

See Also

[SceNpPartyMemberVoiceInfo](#), [sceNpPartyCheckCallback\(\)](#),
[sceNpPartyRegisterHandler\(\)](#)

Error Codes

000004892117

Define Summary

The following error codes can be returned by functions in the NP Party library. Their values are constructed from the NP prefix “0x8055”, followed by the NP Party prefix “25”, and ending with the error number.

Define	Value	Description
SCE_NP_PARTY_ERROR_UNKNOWN	0x80552501	The error is not defined.
SCE_NP_PARTY_ERROR_ALREADY_INITIALIZED	0x80552502	The NP Party library is already initialized.
SCE_NP_PARTY_ERROR_NOT_INITIALIZED	0x80552503	The NP Party library is not initialized.
SCE_NP_PARTY_ERROR_INVALID_ARGUMENT	0x80552504	An inappropriate parameter was supplied to the function.
SCE_NP_PARTY_ERROR_OUT_OF_MEMORY	0x80552505	The operation ran out of memory.
SCE_NP_PARTY_ERROR_NOT_IN_PARTY	0x80552506	The party member is not in a party.
SCE_NP_PARTY_ERROR_VOICE_NOT_ENABLED	0x80552507	The SceNpPartyVoiceEventHandler event handler was not registered.
SCE_NP_PARTY_ERROR_PARTY_MEMBER_NOT_FOUND	0x80552508	The party member with the specified ID was not found.
SCE_NP_PARTY_ERROR_GAME_SESSION_NOT_ENABLED	0x80552509	The SceNpPartyGameSessionEventHandler is not registered.
SCE_NP_PARTY_ERROR_OPERATION_IN_PROGRESS	0x80552510	The operation is in progress.

Other Constants

Party Room Event Types

Define Summary

Define	Value	Description
SCE_NP_PARTY_ROOM_EVENT_JOINED	(0x0001)	Party room event indicating that the current user has joined a party.
SCE_NP_PARTY_ROOM_EVENT_MEMBER_JOINED	(0x0002)	Party room event indicating that a member has joined the party that the current user is in.
SCE_NP_PARTY_ROOM_EVENT_MEMBER_LEFT	(0x0003)	Party room event indicating that the current user has left the party.
SCE_NP_PARTY_ROOM_EVENT_LEFT	(0x0004)	Party room event indicating that a member has left the party that the current user is in.
SCE_NP_PARTY_ROOM_EVENT_MEMBER_PRESENCE_UPDATE	(0x0005)	Party room event indicating that a member's game presence has been updated.

Party Member Voice States

Define Summary

Define	Value	Description
SCE_NP_PARTY_MEMBER_VOICE_STATE_UNKNOWN	(0x0000)	Party voice state indicating that a member's voice peer-to-peer connection is unknown and is being established.
SCE_NP_PARTY_MEMBER_VOICE_STATE_DISCONNECTED	(0x0001)	Party voice state indicating that a member's voice peer-to-peer connection has been disconnected.
SCE_NP_PARTY_MEMBER_VOICE_STATE_CONNECTED	(0x0002)	Party voice state indicating that a member's voice peer-to-peer connection has been established. This event represents the state of a party member that is not currently talking.
SCE_NP_PARTY_MEMBER_VOICE_STATE_TALKING	(0x0003)	Party voice state indicating that a member is talking. There is no corresponding event to indicate when the party member has finished talking; so the game should use a timer to expire a party member from a talking state. It is recommended that games indicate that a party member is in a talking state for no less than 250 ms and no longer than one second after an SCE_NP_PARTY_MEMBER_VOICE_STATE_TALKING voice event occurs. For most games, displaying an SCE_NP_PARTY_MEMBER_VOICE_STATE_TALKING event for one second is recommended.
SCE_NP_PARTY_MEMBER_VOICE_STATE_MUTED	(0x0004)	Party voice state indicating that a remote party member has been muted in the party application.

Party States

Define Summary

Define	Value	Description
SCE_NP_PARTY_STATE_IN_PARTY	(0x0001)	Party room state indicating that the current user is in the party.
SCE_NP_PARTY_STATE_NOT_IN_PARTY	(0x0002)	Party room state indicating that the current user is not in the party.
SCE_NP_PARTY_STATE_IN_PRIVATE_PARTY	(0x0003)	Party room state indicating that the current user is in a private party.

000004892117

Party Room Limits

Define Summary

Define	Value	Description
SCE_NP_PARTY_ROOM_TALKING_LIMIT	1000000	The maximum time to consider a talking state to be valid, in microseconds.
SCE_NP_PARTY_MEMBER_NUM_MAX	8	Maximum number of members in a party.
SCE_NP_PARTY_ID_LENGTH	16	The length limit of the party room ID.

000004892117

Party Member Flags

Define Summary

Define	Value	Description
SCE_NP_PARTY_MEMBER_FLAG_IS_ME	(0x40)	Party member flag indicating that the member is the current user.
SCE_NP_PARTY_MEMBER_FLAG_IS_PARTY_LEADER	(0x80)	Party member flag indicating that the member is the party leader.

000004892117

Party Room Reason for Leaving Flags

Define Summary

Define	Value	Description
SCE_NP_PARTY_ROOM_LEFT_REASON_USER_ACTION	0x00000001	Flag indicating that the party member left due to the member's action, such as quitting the party.
SCE_NP_PARTY_ROOM_LEFT_REASON_SERVER_ERROR	0x00000002	Flag indicating that the party member left due to a server error.
SCE_NP_PARTY_ROOM_LEFT_REASON_KICKED_OUT	0x00000003	Flag indicating that the party member left due to being kicked out of the party.

Party Game-Session Events

Define Summary

Define	Value	Description
SCE_NP_PARTY_GAME_SESSION_UPDATE_EVENT_READY	(0x01)	Party game-session update will be received by the application.
SCE_NP_PARTY_GAME_SESSION_MEMBER_UPDATE_EVENT	(0x02)	Party game-session update was received for a party member. The application should cast the data field into a struct object of type SceNpPartyMemberGameSessionUpdateInfo .

Party Game-Session Update Types

Define Summary

Define	Value	Description
SCE_NP_PARTY_GAME_SESSION_UPDATE_TYPE_NOT_JOINABLE	(0x01)	Party game-session update indicating that a party member is not in a joinable session.
SCE_NP_PARTY_GAME_SESSION_UPDATE_TYPE_JOINABLE	(0x02)	Party game-session update indicating that a party member is in a joinable session.

000004892117