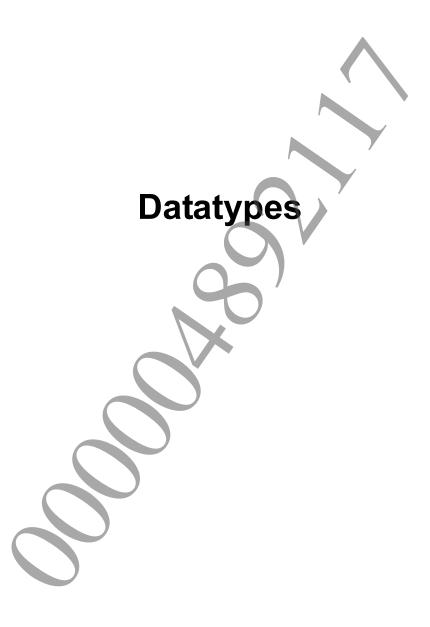


©2015 Sony Computer Entertainment Inc. All Rights Reserved. SCE Confidential

# **Table of Contents**

Datatypes	4
SceNpPartyEventHandlers	5
SceNpPartyGameSessionEvent	6
SceNpPartyGameSessionMemberUpdateType	7
SceNpPartyGameSessionUpdateReady	8
SceNpPartyld	9
SceNpPartyInitParam	10
SceNpPartyMemberFlags	
SceNpPartyMemberGameSessionUpdateInfo	12
SceNpPartyMemberInfo	13
SceNpPartyMemberList	
SceNpPartyMemberVoiceInfo	15
SceNpPartyMemberVoiceState	16
SceNpPartyRoomEventType	17
SceNpPartyRoomLeftReason	
SceNpPartyRoomMemberId	19
SceNpPartyState	20
Functions	21
Library Initialization and Termination	
sceNpPartyInit	
sceNpPartyInitParamInit	23
sceNpPartyTerm	
Event Handling	
sceNpPartyCheckCallback	
sceNpPartyEventHandlersInit	
sceNpPartyRegisterHandler	
Request	
sceNpPartyGetGameSessionReadyState	
sceNpPartyGetId	29
sceNpPartyGetMemberInfo	30
sceNpPartyGetMembers	31
sceNpPartyGetMemberSessionInfo	32
sceNpPartyGetMemberVoiceInfo	33
sceNpPartyGetState	34
Callback Functions	
SceNpPartyGameSessionEventHandler	
SceNpPartyRoomEventHandler	
SceNpPartyVoiceEventHandler	
Error Codes	
Define Summary	
Other Constants	
Party Room Event Types	
Define Summary	42

Party Member Voice States	 43
Define Summary	 43
Party States	
Define Summary	 44
Party Room Limits	 45
Define Summary	
Party Member Flags	
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
= -·······, ···········	



## **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**

sdkVersion SDK version.

roomEventHandler Event handler for subscribing to party room events.

voiceEventHandlerEvent handler for subscribing to party voice notification events.gameSessionEventHandlerEvent 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()

**©SCEI** 

# 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_	If this event is received and the event handler returns
EVENT_READY	SCE OK, the application will receive
	SCE NP PARTY GAME SESSION MEMBER UPDATE
	<b>EVENT</b> events. The data field should be cast to an
	SceNpPartyGameSessionUpdateReady structure.
SCE_NP_PARTY_GAME_SESSION_MEMBER_	The event handler has a game-session update event
UPDATE_EVENT	for a party member. The application should cast the
	data field into a struct object of type
	SceNpPartyMemberGameSessionUpdateInfo.



# **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_	The party member is not in a joinable session
NOT_JOINABLE	with the application using the NP Party
	library
SCE_NP_PARTY_GAME_SESSION_UPDATE_TYPE_	The party member is in a joinable session
JOINABLE	with the application using the NP Party
	library.



## **SceNpPartyGameSessionUpdateReady**

Structure containing the party game session update-ready state.

#### **Definition**

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

#### **Members**

result

If the result is  $SCE\_OK$ , 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



# **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()

## **SceNpPartyInitParam**

Structure containing NP Party initialization parameters.

#### **Definition**

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

#### **Members**

sdkVersion

SDK version.

commId

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()

# **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.



# 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**

type	The game-session update information for the party member specified in
	memberId.
reserved	Reserved field; do not use.
memberId	The member ID of the party member related to this game-session update.
reserved2	Reserved field; do not use.
sessionId	Session ID of the game being played by the party member specified in memberId.
	Only valid when type 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 UPDATE EVENT

### **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**

npId NP ID of the party member. memberId Member ID of the party member. memberFlags Member flags of the party member.

Reserved for internal use of NP Party library. reserved gamePresence Game presence of the party member. Deprecated; use

 ${\tt sceNpPartyGetM\underline{emberSessionInfo} ()} instead of this struct member.$ commId 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.



# **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**

members Array of NP IDs for party members.

memberNum Current count of members in the party.

privateParty Flag that when set to true indicates that only invited party members are allowed

n the party.

padding 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 <a href="SCE\_NP\_PARTY\_MEMBER\_NUM\_MAX">SCE\_NP\_PARTY\_MEMBER\_NUM\_MAX</a> members, where <a href="SCE\_NP\_PARTY\_MEMBER\_NUM\_MAX">SCE\_NP\_PARTY\_MEMBER\_NUM\_MAX</a> is defined as 8.

# SceNpPartyMemberVoiceInfo

Structure containing party member voice information.

#### **Definition**

#### Members

memberId Member ID of the party member.

memberVoiceState Voice state of the party member. One of the states defined by

SceNpPartyMemberVoiceState.

padding Padding to keep the structure 4-byte aligned.

#### **Description**

This structure contains voice information pertaining to a party member.

#### See Also

SceNpPartyMemberVoiceState,SceNpPartyRoomMemberId

# **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
\ <b>X</b>	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.

# **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.



# 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.



# 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.



# **SceNpPartyState**

Type indicating the party room state.

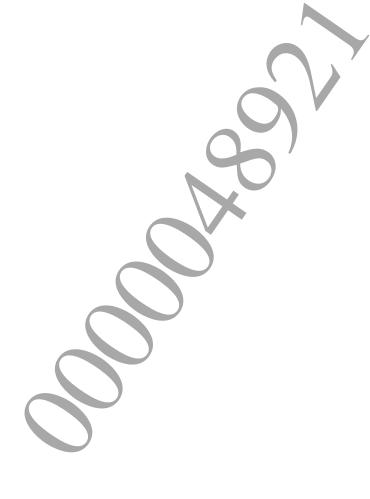
#### **Definition**

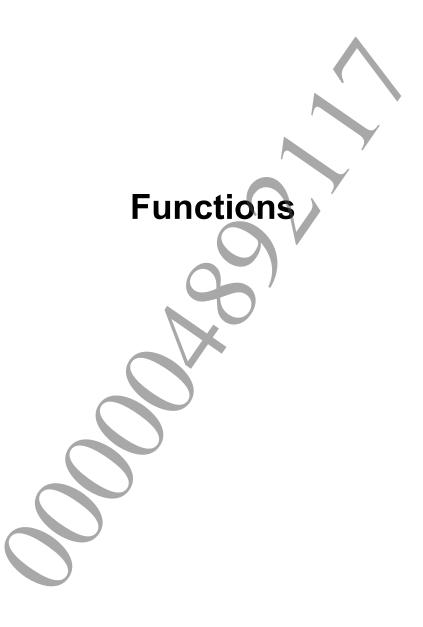
#include <np/np\_party.h>
typedef SceUInt16 SceNpPartyState;

#### **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.





# **Library Initialization and Termination**

## sceNpPartyInit

Initializes the NP Party library.

#### **Definition**

```
#include <np/np_party.h>
int sceNpPartyInit(
    const <u>SceNpPartyInitParam</u> *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 <a href="mailto:sceNpPartyInitParam">sceNpPartyInitParam</a> structure; update the <a href="mailto:sceNpPartyInitParam">SceNpPartyInitParam</a> structure as appropriate for the application; then call <a href="mailto:sceNpPartyInit">sceNpPartyInit</a> () 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**

#### **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 <a href="SceNpPartyInitParam">SceNpPartyInitParam</a> structure; update the <a href="SceNpPartyInitParam">SceNpPartyInitParam</a> structure as appropriate for the application; then call <a href="SceNpPartyInit">SceNpPartyInit</a> () with the updated data.

#### See Also

SceNpPartyInitParam

## 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()

# Document serial number: 000004892117

# **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()

**©SCEI** 

# 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()

## sceNpPartyRegisterHandler

Registers event handlers.

#### **Definition**

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

#### **Calling Conditions**

Not multithread safe.

#### **Arguments**

handlers userdata Structure for the event handler registration. 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

©SCEI

# Request

# sceNpPartyGetGameSessionReadyState

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

#### **Definition**

#### **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_	The SceNpPartyGameSessionEventHandler is
NOT_ENABLED	not registered.
SCE_NP_PARTY_ERROR_OPERATION_IN_	The party game session update-ready system is in
PROGRESS	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

 $\frac{\texttt{sceNpPartyRegisterHandler(),}}{\texttt{SceNpPartyGameSessionEventHandler}}, \\ \frac{\texttt{sceNpPartyGameSessionEventHandler(),}}{\texttt{SceNpPartyGameSessionUpdateReady}}$ 

# sceNpPartyGetId

Gets the party ID.

#### **Definition**

#### **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

**©SCEI** 

## sceNpPartyGetMemberInfo

Retrieves party member details.

#### **Definition**

#### **Calling Conditions**

Not multithread safe.

#### **Arguments**

npId
memberInfo

NP ID of party member.

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
\ <b>X</b>	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()

## sceNpPartyGetMembers

Retrieves the list of current party members.

#### **Definition**

#### **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()

**©SCEI** 

## sceNpPartyGetMemberSessionInfo

Retrieves party member game-session details.

#### **Definition**

```
#include <np/np_party.h>
int sceNpPartyGetMemberSessionInfo(
    const <u>SceNpPartyRoomMemberId</u> memberId,
        <u>SceNpPartyMemberGameSessionUpdateInfo</u>* 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	SceNpPartyGameSessionEventHandler
ENABLED	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 <a href="SceNpPartyRoomMemberId">SceNpPartyRoomMemberId</a>. 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

 $\frac{\texttt{SceNpPartyMemberGameSessionUpdateInfo}}{\texttt{SceNpPartyGameSessionEventHandler}}, \frac{\texttt{SceNpPartyRoomMemberId}}{\texttt{SceNpPartyGameSessionEventHandler}}$ 

## 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
memberVoiceState

<u>SceNpPartyRoomMemberId</u> of party member. 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

## sceNpPartyGetState

Retrieves the party state of the current user.

#### **Definition**

#### **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.

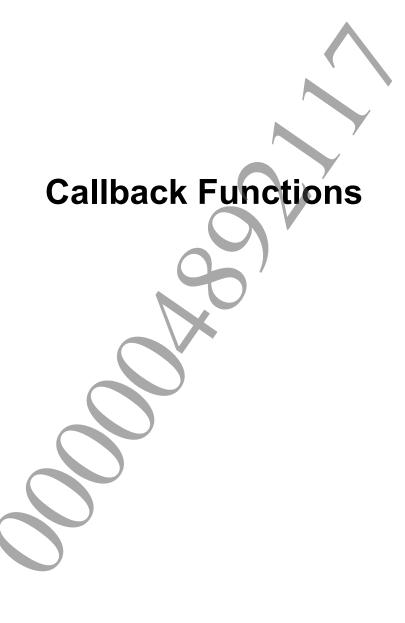
#### **Notes**

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

**©SCEI** 



## **SceNpPartyGameSessionEventHandler**

Event handler for receiving party member game-session updates.

#### **Definition**

#### **Arguments**

event	The type of event being returned. If event is
	SCE_NP_PARTY_GAME_SESSION_UPDATE_EVENT_READY, the data field should
	be cast to <a href="SceNpPartyGameSessionUpdateReady">SceNpPartyGameSessionUpdateReady</a> . If event is
	SCE_NP_PARTY_GAME_SESSION_MEMBER_UPDATE_EVENT, the data field
	should be cast to <a href="SceNpPartyMemberGameSessionUpdateInfo">SceNpPartyMemberGameSessionUpdateInfo</a> .
data	The data field should be cast to the appropriate type as specified by event.
userdata	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

 $\frac{\texttt{SceNpPartyMemberGameSessionUpdateInfo,}}{\texttt{sceNpPartyCheckCallback(),}} \underbrace{\frac{\texttt{SceNpPartyCheckCallback(),}}{\texttt{sceNpPartyGameSessionEvent}}} \underbrace{\texttt{sceNpPartyRegisterHandler(),}} \\$ 

### **SceNpPartyRoomEventHandler**

Event handler for receiving party room events.

#### **Definition**

### **Arguments**

eventType Generated event type.

data Event data.

userdata 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 <a href="mailto:scenpPartyRegisterHandler">scenpPartyRegisterHandler</a>(), NP Party generates events for notifying the application of various party room events as described by <a href="mailto:scenpPartyRoomEventType">ScenpPartyRoomEventType</a>. When the application calls <a href="mailto:scenpPartyCheckCallback">scenpPartyCheckCallback</a>() 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()

### **SceNpPartyVoiceEventHandler**

Event handler for receiving party voice notification events.

#### **Definition**

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

### **Arguments**

memberVoiceInfo
userdata

Generated event type.

User data set upon registering the event handler.

#### **Return Values**

None

### **Description**

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

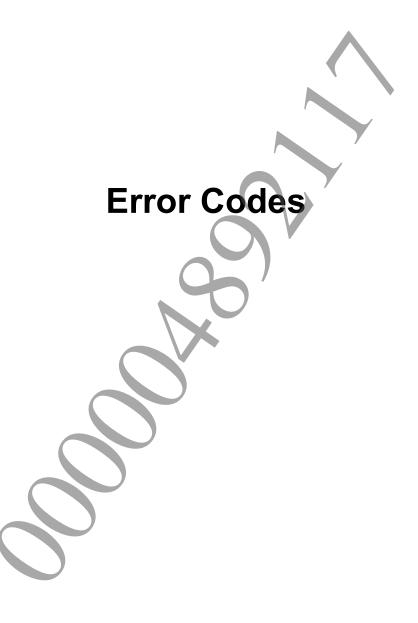
When the event handler is registered using <a href="sceNpPartyRegisterHandler">sceNpPartyRegisterHandler</a>(), NP Party generates voice notification events for notifying the application of when various members of the party are speaking. When the application calls <a href="sceNpPartyCheckCallback">sceNpPartyCheckCallback</a>() 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

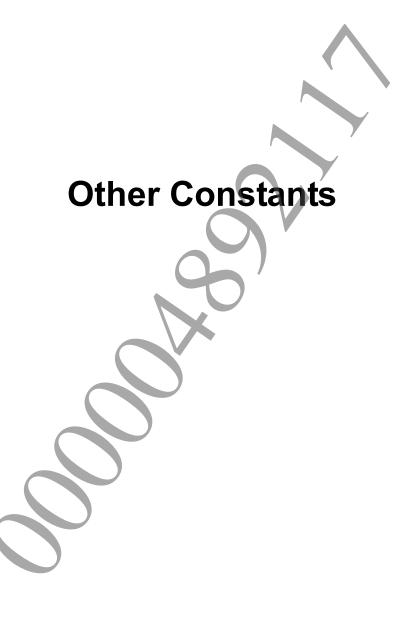
SceNpPartyMemberVoiceInfo, sceNpPartyCheckCallback(),
sceNpPartyRegisterHandler()



### **Define Summary**

The following error codes can be returned by functions in the NP Party library. Their values are constructed from the NP prefix " $0 \times 8055$ ", 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_	0x80552502	The NP Party library is already initialized.
INITIALIZED		
SCE_NP_PARTY_ERROR_NOT_	0x80552503	The NP Party library is not initialized.
INITIALIZED		
SCE_NP_PARTY_ERROR_INVALID_	0x80552504	An inappropriate parameter was supplied to the
ARGUMENT		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_	0x80552507	The SceNpPartyVoiceEventHandler event
ENABLED		handler was not registered.
SCE_NP_PARTY_ERROR_PARTY_MEMBER_	0x80552508	The party member with the specified ID was not
NOT_FOUND		found.
SCE_NP_PARTY_ERROR_GAME_SESSION_	0x80552509	The SceNpPartyGameSessionEventHandler
NOT_ENABLED		is not registered.
SCE_NP_PARTY_ERROR_OPERATION_IN_	0x80552510	The operation is in progress.
PROGRESS		



# **Party Room Event Types**

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
	7	has been updated.

# **Party Member Voice States**

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
	ſ	games indicate that a party
\ <b>X</b>		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.

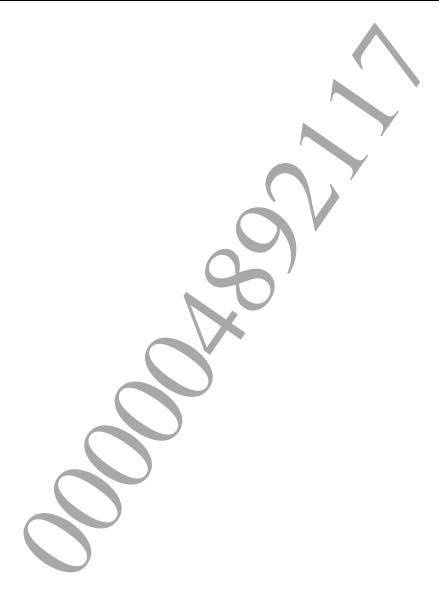
# **Party States**

Define	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.



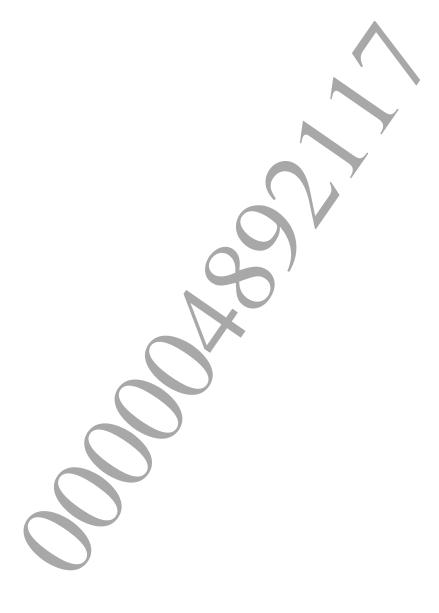
# **Party Room Limits**

Define		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.



# **Party Member Flags**

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.



# **Party Room Reason for Leaving Flags**

Define	Value	Description
SCE_NP_PARTY_ROOM_LEFT_REASON_USER_ACTION	0x0000001	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	0x0000003	Flag indicating that the party
		member left due to being kicked
		out of the party.



# **Party Game-Session Events**

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	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.

