NP Library Reference

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Document serial number: 000004892117

Document serial number: 000004892117

Table of Contents

Initialization/Termination

SceNpCommunicationConfig

Configurations for PSN℠

Definition

```
#include <np.h>
typedef struct SceNpCommunicationConfig{
        const SceNpCommunicationId *comId;
        const SceNpCommunicationPassphrase *commPassphrase;
        const SceNpCommunicationSignature *commSignature;
} SceNpCommunicationConfig;
```

Members

comId	NP Communication ID (specify the value issued by the PlayStation®Vita Developer Network)
commPassphrase	NP communication passphrase (specify the value issued by the PlayStation®Vita Developer Network)
commSignature	NP communication signature (specify the value issued by the PlayStation®Vita Developer Network)

Description

This structure holds various settings for PSN℠; it is passed as an argument of `sceNpInit()`.

For each member, specify the appropriate value issued by the PlayStation®Vita Developer Network (https://psvita.scedev.net/) after registration.

See Also

`sceNpInit()`

SceNpCommunicationId

NP Communication ID

Definition

```
#include <np.h>
typedef struct SceNpCommunicationId {
        char data[9];
        char term;
        SceUChar8 num;
        char dummy;
} SceNpCommunicationId;
```

Members

data	NP Communication ID string (4 alphabet characters and 5 numeric characters)
term	String terminator area to end *data*
num	Sub ID (usually 0)
dummy	(not used)

Description

This structure represents the NP Communication ID. Store the values issued by the PlayStation®Vita Developer Network after registration in this structure and specify it as a member of `SceNpCommunicationConfig`.

See Also

`sceNpInit(),SceNpCommunicationConfig`

SceNpCommunicationPassphrase

NP communication passphrase

Definition

```
#include <np.h>
#define SCE_NP_COMMUNICATION_PASSPHRASE_SIZE        (128)
typedef struct SceNpCommunicationPassphrase {
        SceUChar8 data[SCE_NP_COMMUNICATION_PASSPHRASE_SIZE];
} SceNpCommunicationPassphrase;
```

Members

data NP communication passphrase
(specify the value issued by the PlayStation®Vita Developer Network)

Description

This is the NP communication passphrase used by the server of PSN℠. Store the value issued by the PlayStation®Vita Developer Network after registration in this structure and specify it as a member of `SceNpCommunicationConfig`.

See Also

`SceNpCommunicationConfig`, `sceNpInit()`

SceNpCommunicationSignature

NP communication signature

Definition

```
#include <np.h>
#define SCE_NP_COMMUNICATION_SIGNATURE_SIZE        (160)
typedef struct SceNpCommunicationSignature {
        SceUChar8 data[SCE_NP_COMMUNICATION_SIGNATURE_SIZE];
} SceNpCommunicationSignature;
```

Members

data NP communication signature data
(specify the value issue by the PlayStation®Vita Developer Network)

Description

This is the signature data used by the server of PSN℠. Store the value issued by the PlayStation®Vita Developer Network after registration in this structure and specify it as a member of `SceNpCommunicationConfig`.

See Also

`SceNpCommunicationConfig`, `sceNpInit()`

SceNpOptParam

Optional parameter

Definition

```
#include <np.h>
typedef struct SceNpOptParam {
        SceSize optParamSize;
} SceNpOptParam;
```

Members

optParamSize Size of this structure (bytes)

Description

This is an optional parameter to be passed as an argument to `sceNpInit()`.
This argument is reserved for future extension and cannot be used in the current SDK.

See Also

`sceNpInit()`

sceNpInit

Initialize the NP library

Definition

```
#include <np.h>
int sceNpInit(
        const SceNpCommunicationConfig *commConf,
        SceNpOptParam *opt
);
```

Arguments

commConf	Configurations for PSN℠. Specify the values issued by the PlayStation®Vita Developer Network. If no values have been issued, NULL can be specified.
opt	Optional parameter. Specify NULL.

Return Values

Returns 0 for normal termination.

Returns an error code upon error.

Description

This function initializes the NP library.

opt is an argument reserved for future extension and cannot be used in the current SDK (specify NULL).

Notes

This function is not multithread safe. Calling this function from multiple threads at the same time will lead to undefined operation.

Example

```
int ret = 0;

/* s_npCommunicationConfig is
 * the variable of SceNpCommunicationConfig structure
 * issued by PlayStation®Vita Developer Network.
 */
ret = sceNpInit( & s_npCommunicationConfig, NULL );
if(ret < 0){
        /* Error handling */
}
```

See Also

SceNpCommunicationConfig, sceNpTerm()

sceNpTerm

Terminate the NP library

Definition

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

Arguments

None

Return Values

None

Description

This function terminates the NP library.

Notes

This function is not multithread safe. Calling this function from multiple threads at the same time will lead to undefined operation.

Example

```
sceNpTerm();
```

See Also

```
sceNpInit()
```

Obtaining Information

SceNpOnlineId

Online ID

Definition

```
#include <np/np_common.h>
#define SCE_NP_ONLINEID_MIN_LENGTH         3
#define SCE_NP_ONLINEID_MAX_LENGTH         16
typedef struct SceNpOnlineId {
        char data[SCE_NP_ONLINEID_MAX_LENGTH];
        char term;
        char dummy[3];
} SceNpOnlineId;
```

Members

data	Online ID character string
term	String terminator area to end *data*
dummy	(Not used)

Description

This structure represents a user's Online ID.

SceNpId

NP ID

Definition

```
#include <np/np_common.h>
typedef struct SceNpId {
        SceNpOnlineId handle;
        SceUChar8 opt[8];
        SceUChar8 reserved[8];
} SceNpId;
```

Members

handle	Online ID
opt	Option data
reserved	Area reserved for future extension

Description

This structure represents the NP ID to be used by the NP library to identify a user.
It internally holds the Online ID.

SceNpCountryCode

Country/region code

Definition

```
#include <np/np_common.h>
typedef struct SceNpCountryCode {
        char data[2];
        char term;
        char padding[1];
} SceNpCountryCode;
```

Members

data	Country/region code (ISO 3166-1)
term	Area for termination character (for terminating *data*)
padding	Padding

Description

This structure represents a country/region. *data* is a 2-byte ASCII code representing a country/region. The country/region code is defined in ISO 3166-1.

The countries/regions currently supported by PSN™ are as follows. It is possible for the list of supported countries/regions to change at any time (and not necessarily in tandem with SDK updates). The application must be programmed so that it will not hang even if an unknown country/region code (a code not in the list below) is obtained.

data	Country/Region
ae	UAE
ar	Argentina
at	Austria
au	Australia
be	Belgium
bg	Bulgaria
bh	Bahrain
br	Brazil
ca	Canada
ch	Switzerland
cl	Chile
co	Colombia
cy	Cyprus
cz	Czech Republic
de	Germany
dk	Denmark
es	Spain
fi	Finland
fr	France
gb	UK
gr	Greece
hk	Hong Kong
hr	Croatia
hu	Hungary
id	Indonesia
ie	Ireland

data	Country/Region
il	Israel
in	India
is	Iceland
it	Italy
jp	Japan
kr	Korea
kw	Kuwait
lb	Lebanon
lu	Luxembourg
mt	Malta
mx	Mexico
my	Malaysia
nl	Netherlands
no	Norway
nz	New Zealand
om	Oman
pe	Peru
pl	Poland
pt	Portugal
qa	Qatar
ro	Romania
ru	Russia
sa	Saudi Arabia
se	Sweden
sg	Singapore
si	Slovenia
sk	Slovakia
th	Thailand
tr	Turkey
tw	Taiwan, Province of China
ua	Ukraine
us	United States
za	South Africa

See Also

`sceNpManagerGetAccountRegion(),SCE_NP_SUBJECT_REGION_GET_COUNTRY_CODE`

sceNpManagerGetNpId

Get own user ID

Definition

```
#include <np/np_manager.h>
int sceNpManagerGetNpId(
                SceNpId *npId
);
```

Arguments

npId	Where to store the obtained NP ID

Return Values

Returns 0 for normal termination.

Returns an error code upon error.

Description

This function obtains the user's own NP ID. The NP ID is saved inside the system when Network Check Dialog is used in the PSN℠ mode. If this function is called before calling Network Check Dialog, `SCE_NP_MANAGER_ERROR_NEED_CALL_NETCHECK_DIALOG` is returned.

Example

```
int ret = 0;
SceNpId npId;

ret = sceNpManagerGetNpId(&npId);
if(ret < 0){
        /* Error handling */
}
```

sceNpManagerGetAccountRegion

Get region information

Definition

```
#include <np/np_manager.h>
int sceNpManagerGetAccountRegion(
        SceNpCountryCode *countryCode,
        int *language
);
```

Arguments

countryCode	Country/region code
language	Language code

Return Values

Returns 0 for normal termination.

Returns an error code upon error.

Description

This function gets the user's region information and stores it to **countryCode* and **language*.

For an explanation of the values stored to **countryCode*, refer to the description of SceNpCountryCode.

The value stored to **language* is as follows

Value	(Number)	Description
SCE_NP_LANG_JAPANESE	0	Japanese
SCE_NP_LANG_ENGLISH	1	English (United States)
SCE_NP_LANG_FRENCH	2	French
SCE_NP_LANG_SPANISH	3	Spanish
SCE_NP_LANG_GERMAN	4	German
SCE_NP_LANG_ITALIAN	5	Italian
SCE_NP_LANG_DUTCH	6	Dutch
SCE_NP_LANG_PORTUGUESE	7	Portuguese (Portugal)
SCE_NP_LANG_RUSSIAN	8	Russian
SCE_NP_LANG_KOREAN	9	Korean
SCE_NP_LANG_CHINESE_T	10	Chinese (traditional)
SCE_NP_LANG_CHINESE_S	11	Chinese (simplified)
SCE_NP_LANG_FINNISH	12	Finnish
SCE_NP_LANG_SWEDISH	13	Swedish
SCE_NP_LANG_DANISH	14	Danish
SCE_NP_LANG_NORWEGIAN	15	Norwegian
SCE_NP_LANG_POLISH	16	Polish
SCE_NP_LANG_PORTUGUESE_BR	17	Portuguese (Brazil)
SCE_NP_LANG_ENGLISH_GB	18	English (United Kingdom)
SCE_NP_LANG_TURKISH	19	Turkish

The information that can be obtained with this function is the information that was obtained upon signing in to the PSN℠ server and held within the utility. The support state of country/region and language codes may be added to/revised any time irrespective of SDK updates. Program the application so that it does not break down even if a country/region code or language code that is not indicated here is obtained.

If this function is called before calling Network Check Dialog, `SCE_NP_MANAGER_ERROR_NEED_CALL_NETCHECK_DIALOG` may be returned.

Examples

```
int ret = 0;
SceNpCountryCode countryCode;
int language;

ret = sceNpManagerGetAccountRegion(&countryCode, language);
if(ret < 0){
        /* Error handling */
}
```

sceNpManagerGetContentRatingFlag

Get parental controls information (age-based content restriction) (function for preserving compatibility)

Definition

```
#include <np/np_manager.h>
int sceNpManagerGetContentRatingFlag(
        int *isRestricted,
        int *age
);
```

Arguments

isRestricted	Restriction flag (True if parental controls settings are set)
age	User's age

Return Values

Returns 0 for normal termination.

Returns an error code upon error.

Description

This function is provided for preserving compatibility. Checking content restriction is performed internally by Network Check Dialog, therefore there is no need for applications to call this function.

If this function is called before calling Network Check Dialog, `SCE_NP_MANAGER_ERROR_NEED_CALL_NETCHECK_DIALOG` may be returned.

sceNpManagerGetChatRestrictionFlag

Get parental controls information (chat restrictions)

Definition

```
#include <np/np_manager.h>
int sceNpManagerGetChatRestrictionFlag(
        int *isRestricted
);
```

Arguments

isRestricted Restriction flag (True if parental controls settings are set)

Return Values

Returns 0 for normal termination.

Returns an error code upon error.

Description

This function gets the user's parental controls information and stores it to **isRestricted*.

If the value of **isRestricted* is True, it is necessary to restrict online communication services for the user, including text chats, audio/video chats, and messaging.

This information can be found inside the utility, where it was stored after being obtained from the server of PSN℠ when the user signed in to PSN℠.

If this function is called before calling Network Check Dialog, `SCE_NP_MANAGER_ERROR_NEED_CALL_NETCHECK_DIALOG` may be returned.

Examples

```
int ret = 0;
int isRestricted;

ret = sceNpManagerGetChatRestrictionFlag(&isRestricted);
if(ret < 0){
        /* Error handling */
}
```

sceNpManagerGetCachedParam

Get NP information stored in flash

Definition

```
#include <np/np_manager.h>
int sceNpManagerGetCachedParam(
        SceNpManagerCacheParam *param
);
```

Arguments

param NP information stored in flash

Return Values

Returns 0 for normal termination.

Returns an error code upon error.

Description

This function gets the user's NP ID and Avatar URL and stores them into **param*.

`SceNpManagerCacheParam` has the following members.

```
typedef struct SceNpManagerCacheParam {
        SceNpId npId;
        SceNpAvatarUrl avatarUrl;
} SceNpManagerCacheParam;
```

The information acquired with this function is obtained from the server of PSN℠ and stored in an internal flash at the following timing.

- When executing sign-in processing

Examples

```
int ret = 0;
SceNpManagerCacheParam param;

ret = sceNpManagerGetCachedParam(&param);
if(ret < 0){
        /* Error handling */
}
```

Obtaining/Monitoring the Service State

SceNpServiceState

Service state

Definition

```
#include <np.h>
enum SceNpServiceState {
        SCE_NP_SERVICE_STATE_UNKNOWN = 0,
        SCE_NP_SERVICE_STATE_SIGNED_OUT,
        SCE_NP_SERVICE_STATE_SIGNED_IN,
        SCE_NP_SERVICE_STATE_ONLINE
};
```

Enumeration Value

Value	(Number)	Description
SCE_NP_SERVICE_STATE_UNKNOWN	0	Unknown state
SCE_NP_SERVICE_STATE_SIGNED_OUT	1	Signed-out. State in which most of PSN℠ features cannot be used.
SCE_NP_SERVICE_STATE_SIGNED_IN	2	Signed-in. State in which PSN℠ features can be used.
SCE_NP_SERVICE_STATE_ONLINE	3	Online. State in which PSN℠ features can be used and real-time communication is possible.

Description

This enumeration type represents the service state of the NP library.

Depending on the service state, usable PSN℠ features differ, as follows.

SCE_NP_SERVICE_STATE_SIGNED_OUT represents a state in which most of PSN℠ features cannot be used.

SCE_NP_SERVICE_STATE_SIGNED_IN represents a state in which PSN℠ services can be used by using tickets.

SCE_NP_SERVICE_STATE_ONLINE represents a state in which PSN℠ services can be used by using tickets and the real-time communication feature (made possible by the NP Basic library) can also be used.

See Also

sceNpGetServiceState(),SceNpServiceStateCallback

SceNpServiceStateCallback

Callback of the service state

Definition

```
#include <np.h>
typedef void (*SceNpServiceStateCallback)(
        SceNpServiceState state,
        int retCode,
        void *userdata
);
```

Arguments

state	Service state
retCode	Return code
userdata	User data set upon registering the callback

Return Values

None

Description

This is a prototype for the callback function for receiving notifications regarding changes in the service state.

When registering a callback function with `sceNpRegisterServiceStateCallback()`, an event notifying the service state at that point will be generated. Subsequently, the registered callback function will be called whenever the application calls `sceNpCheckCallback()`. The *state* argument represents the service state; perform appropriate processing in accordance to this value.

Henceforth, an event notifying the new service state will be generated every time the service state changes according to network connect/disconnection or a user operation. When the application calls `sceNpCheckCallback()` after an event generation, this callback function will be called and the new service state will be passed to the *state* argument.

Depending on the service state, usable PSN℠ features differ. If the service state is not appropriate for the feature you want to use (for example, if real-time communication feature of the NP Basic library cannot be used because the service state is signed-in), use Network Check Dialog to prompt the user to switch to the appropriate service state.

If the change in service state is due to an internal system error, the error code for the corresponding error will be set to *retCode*.

Notes

Avoid processing within this callback that takes a long period of time to complete and ensure that the callback function returns immediately.

See Also

`SceNpServiceState`, `sceNpRegisterServiceStateCallback()`

sceNpGetServiceState

Obtain the service state

Definition

```
#include <np.h>
int sceNpGetServiceState(
        SceNpServiceState *state
);
```

Arguments

state Where to store the obtained service state

Return Values

Returns 0 for normal termination.

Returns an error code upon error.

Description

This function obtains the service state.

Notes

This function is multithread safe.

Because this function entails communication between processes on the system process side in order to request processing to the system process, this function can block other processes for a long period of time depending on the system process load. Do not call this function from a thread onto which you do not want the effects of the system process load (the rendering thread, for example).

sceNpRegisterServiceStateCallback

Register the callback function for receiving the service state

Definition

```
#include <np.h>
int sceNpRegisterServiceStateCallback(
        SceNpServiceStateCallback callback,
        void *userdata
);
```

Arguments

callback	Callback function
userdata	Arbitrary user data to pass to the callback function

Return Values

Returns 0 for normal termination.

Returns an error code upon error.

Description

This function registers the callback function for receiving notifications regarding changes of the service state.

Notes

This function is not multithread safe.

Because this function entails communication between processes on the system process side in order to request processing to the system process, this function can block other processes for a long period of time depending on the system process load. Do not call this function from a thread onto which you do not want the effects of the system process load (the rendering thread, for example).

See Also

SceNpServiceStateCallback

sceNpUnregisterServiceStateCallback

Unregister the callback function for receiving the service state

Definition

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

Arguments

None

Return Values

Returns 0 for normal termination.

Returns an error code upon error.

Description

This function unregisters the callback function for receiving notifications regarding changes in the service state.

Notes

This function is not multithread safe.

Because this function entails communication between processes on the system process side in order to request processing to the system process, this function can block other processes for a long period of time depending on the system process load. Do not call this function from a thread onto which you do not want the effects of the system process load (the rendering thread, for example).

See Also

`sceNpRegisterServiceStateCallback()`

General Function

sceNpCheckCallback

Check callback functions

Definition

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

Arguments

None

Return Values

Returns 0 for normal termination.

Returns an error code upon error.

Description

This function checks whether any of the callback functions or event handlers registered using the NP library or other libraries related to PSN℠, or the callback for receiving the result of an asynchronous API, are in a state in which they should be called. If so - in other words, if an applicable event has been generated or if processing within the asynchronous API has completed and its result is ready to be returned to the application - the context of the thread that called this function will call the appropriate callback function.

Design the application so that this function is regularly called between the call of `sceNpInit()` and `sceNpTerm()`.

Notes

This function is not multithread safe.

Out of all the libraries related to PSN℠, some libraries have a function unique to the respective library that performs equivalent processing to this function.

Utility Functions

sceNpCmpNpId

Compare NP IDs

Definition

```
#include <np/np_common.h>
int sceNpCmpNpId (
        const SceNpId *npid1,
        const SceNpId *npid2
);
```

Arguments

npid1	NP ID(1) for comparison
npid2	NP ID(2) for comparison

Return Values

Returns 0 when the two NP IDs match.

Returns `SCE_NP_UTIL_ERROR_NOT_MATCH` if the two NP IDs do not match.

If there is an error, returns the error code without performing the comparison.

Description

This function compares *npid1* and *npid2*. It returns 0 if they match.

When comparing NP IDs to see if they match, be sure to use this function instead of directly comparing the NP ID structures.

sceNpCmpNpIdInOrder

Compare NP IDs (with order)

Definition

```
#include <np/np_common.h>
int sceNpCmpNpIdInOrder (
        const SceNpId *npid1,
        const SceNpId *npid2,
        int *order
);
```

Arguments

npid1	NP ID(1) for comparison
npid2	NP ID(2) for comparison
order	Comparison result

Return Values

For normal termination, stores the comparison result to *order* and returns 0.

Returns an error code upon error.

Description

This function compares *npid1* and *npid2* and obtains a unique order.

If the two NP IDs match, it saves 0 to *order*, and if they do not match, it returns either a positive or a negative value determined uniquely between the two NP IDs. The value returned to *order* is equal to the result obtained by executing `memcmp()` for the online ID part of the NP IDs.

sceNpCmpOnlineId

Compare NP IDs (online IDs only)

Definition

```
#include <np/np_common.h>
int sceNpCmpOnlineId (
        const SceNpId *npid1,
        const SceNpId *npid2
);
```

Arguments

npid1	NP ID(1) for comparison
npid2	NP ID(2) for comparison

Return Values

Returns 0 when the online IDs included in the two NP IDs match.

Returns `SCE_NP_UTIL_ERROR_NOT_MATCH` if they do not match.

If there is an error, returns the error code without performing the comparison.

Description

This function compares *npid1* and *npid2*, comparing the online IDs included in the NP IDs. It returns 0 if they match.

This function ignores the platform information included in the NP IDs and compares only the users' online IDs.

sceNpGetPlatformType

Get platform from NP ID

Definition

```
#include <np/np_common.h>
int sceNpGetPlatformType (
        const SceNpId *npId
);
```

Arguments

npId NP ID

Return Values

Returns a platform type of 0 or greater for normal termination.
Returns an error code upon error.

Description

This function returns the platform type included in *npId*.

See Also

`SceNpPlatformType` (in the "Platform Types" section)

sceNpSetPlatformType

Set the platform to NP ID

Definition

```
#include <np/np_common.h>
int sceNpSetPlatformType (
        SceNpId *npId,
        SceNpPlatformType platformType
);
```

Arguments

npId	NP ID
platformType	Platform type set to *npId*

Return Values

Returns 0 for normal termination.

Returns an error code upon error.

Description

Sets the value of *platformType* to *npId*.

See Also

SceNpPlatformType (in the "Platform Types" section)

User Information Structures

SceNpAvatarUrl

Structure for Avatar

Definition

```
#include <np/np_common.h>
typedef struct SceNpAvatarUrl {
        char data[SCE_NP_AVATAR_URL_MAX_LENGTH];
        char term;
} SceNpAvatarUrl;
```

Members

data	URL indicating Avatar data
term	Area for termination character (for terminating *data*)

Description

This structure represents the Avatar of the user.

SceNpUserInformation

Structure for user information

Definition

```
#include <np/np_common.h>
typedef struct SceNpUserInformation {
        SceNpId userId;
        SceNpAvatarUrl icon;
        SceUChar8 reserved[52];
} SceNpUserInformation;
```

Members

userId	NP ID
icon	Avatar
reserved	Unused area

Description

This structure contains information indicating the user. It consists of the identifier NP ID, and the additional information of the Avatar URL.

SceNpMyLanguages

Structure for languages used often

Definition

```
#include <np/np_common.h>
struct SceNpMyLanguages{
        SceInt32 language1;
        SceInt32 language2;
        SceInt32 language3;
        SceUChar8 padding[4];
} SceNpMyLanguages;
```

Members

language1	Language setting 1 (always set)
language2	Language setting 2 (negative when not set)
language3	Language setting 3 (negative when not set)
padding	Padding

Description

This structure has information of the "languages often used" as set by the user.

The following values can be set to *language1*, *language2*, and *language3*. If there is no value set to a member, a negative value will be set.

Value	(Number)	Description
SCE_NP_LANG_JAPANESE	0	Japanese
SCE_NP_LANG_ENGLISH_US	1	English (United States)
SCE_NP_LANG_FRENCH	2	French (France)
SCE_NP_LANG_SPANISH	3	Spanish (Spain)
SCE_NP_LANG_GERMAN	4	German
SCE_NP_LANG_ITALIAN	5	Italian
SCE_NP_LANG_DUTCH	6	Dutch
SCE_NP_LANG_PORTUGUESE_PT	7	Portuguese (Portugal)
SCE_NP_LANG_RUSSIAN	8	Russian
SCE_NP_LANG_KOREAN	9	Korean
SCE_NP_LANG_CHINESE_T	10	Chinese (traditional)
SCE_NP_LANG_CHINESE_S	11	Chinese (simplified)
SCE_NP_LANG_FINNISH	12	Finnish
SCE_NP_LANG_SWEDISH	13	Swedish
SCE_NP_LANG_DANISH	14	Danish
SCE_NP_LANG_NORWEGIAN	15	Norwegian
SCE_NP_LANG_POLISH	16	Polish
SCE_NP_LANG_PORTUGUESE_BR	17	Portuguese (Brazil)
SCE_NP_LANG_ENGLISH_GB	18	English (United Kingdom)
SCE_NP_LANG_TURKISH	19	Turkish
SCE_NP_LANG_SPANISH_LA	20	Spanish (Latin America)
SCE_NP_LANG_ARABIC	21	Arabic
SCE_NP_LANG_FRENCH_CA	22	French (Canada)

SceNpAvatarImage

Structure for the Avatar image

Definition

```
#include <np/np_common.h>
typedef struct SceNpAvatarImage{
        SceUChar8 data[SCE_NP_AVATAR_IMAGE_MAX_SIZE];
        SceSize size;
        SceUChar8 reserved[12]
} SceNpAvatarImage;
```

Members

data	Avatar image data
size	Size of *data*
reserved	Reserved for future extension

Description

This structure is for storing an Avatar image.

There are three types of Avatar images - SCE_NP_AVATAR_SIZE_LARGE, SCE_NP_AVATAR_SIZE_MIDDLE, and SCE_NP_AVATAR_SIZE_SMALL.

Value	Description
SCE_NP_AVATAR_SIZE_LARGE	Large-sized Avatar image The format is a 240 x 240 px png, and RGBA is 8 bits each. The maximum size is 200KB.
SCE_NP_AVATAR_SIZE_MIDDLE	Middle-sized Avatar image The format is a 160 x 160 px png, and RGBA is 8 bits each and non-interlaced. The maximum size is 100KB.
SCE_NP_AVATAR_SIZE_SMALL	Small-sized Avatar image The format is a 50 x 50 px png, and RGBA is 8 bits each and non-interlaced. The maximum size is 10KB.

Notes

Because of the large data size, take care not to allocate the data to an automatic variable.

SceNpAboutMe

Structure for the self introduction

Definition

```
#include <np.h>
typedef struct SceNpAboutMe{
        char data[SCE_NP_ABOUT_ME_MAX_LENGTH + 1];
} SceNpAboutMe;
```

Members

data Self introduction string (UTF-8)

Description

This structure represents a user's introductory note.

PSN[SM] Web API-related

SceNpTitleId

NP Title ID

Definition

```
#include <np/np_common.h>
#define SCE_NP_TITLE_ID_LEN 12
typedef struct SceNpTitleId {
        char id[SCE_NP_TITLE_ID_LEN + 1];
        SceUChar8 padding[3];
} SceNpTitleId;
```

Members

id	NP Title ID string (specify the value issued by the PlayStation®Vita Developer Network)
padding	Not used (fill with all 0s)

Description

This structure represents the NP Title ID.

To have an NP Title ID issued, make a request through the PlayStation®Vita Developer Network website.

SceNpTitleSecret

NP Title Secret

Definition

```
#include <np/np_common.h>
#define SCE_NP_TITLE_SECRET_SIZE 128
typedef struct SceNpTitleSecret {
        uint8 data[SCE_NP_TITLE_SECRET_SIZE];
} SceNpTitleSecret;
```

Members

data	NP Title Secret data (specify the value issued by the PlayStation®Vita Developer Network)

Description

This structure represents the NP Title Secret.

To issue an NP Title Secret, make a request through the PlayStation®Vita Developer Network website.

SceNpServiceLabel

NP service label

Definition

```
#include <np/np_common.h>
typedef SceUInt32 SceNpServiceLabel;
#define SCE_NP_DEFAULT_SERVICE_LABEL (0x00000000)
#define SCE_NP_INVALID_SERVICE_LABEL (0xFFFFFFFF)
```

Description

This integer indicates the NP service label.

An NP service label is an identifier that specifies when an NP title ID handles multiple service instances. It is a 1-12 digit integer expressed in decimal format. NP service labels are allocated by SCE.

To issue an NP service label, make a request through the PlayStation®Vita Developer Network.

See Also

SceNpCommunicationId

SceNpClientId

Client ID

Definition

```
#include <np/np_common.h>
#define SCE_NP_CLIENT_ID_MAX_LEN 128
typedef struct SceNpClientId {
        char id[SCE_NP_CLIENT_ID_MAX_LEN + 1];
        SceUInt8 padding[7];
} SceNpClientId;
```

Members

id	Client ID string (specify the value issued by the PlayStation®Vita Developer Network)
padding	Not used (fill with all 0s)

Description

This structure represents the client ID.

For details on client IDs, refer to "PSN℠ Web APIs Overview" document.

SceNpAuthorizationCode

Authorization code

Definition

```
#include <np/np_common.h>
#define SCE_NP_AUTHORIZATION_CODE_MAX_LEN 128
typedef struct SceNpAuthorizationCode {
        char code[SCE_NP_AUTHORIZATION_CODE_MAX_LEN + 1];
        SceUInt8 padding[7];
} SceNpAuthorizationCode;
```

Members

code	Authorization code string
padding	Not used

Description

This structure represents the authorization code.

For details on authorization codes, refer to "NP Auth Library Overview" document.

SceNpPeerAddress

Peer address

Definition

```
#include <np/np_common.h>
typedef struct SceNpPeerAddress {
        SceNpOnlineId onlineId;
        SceNpPlatformType platform;
} SceNpPeerAddress;
```

Members

onlineId	Online ID
platform	Platform type

Description

This structure represents the peer address.

Constants

Data Sizes

Data sizes used by the NP client libraries

Definition

Value	(Number)	Description
SCE_NP_AVATAR_IMAGE_MAX_SIZE	200*1024	Maximum size of the Avatar image
SCE_NP_AVATAR_IMAGE_MAX_SIZE_LARGE	200*1024	Maximum size of the (large) Avatar image
SCE_NP_AVATAR_IMAGE_MAX_SIZE_MIDDLE	100*1024	Maximum size of the (middle) Avatar image
SCE_NP_AVATAR_IMAGE_MAX_SIZE_SMALL	10*1024	Maximum size of the (small) Avatar image
SCE_NP_AVATAR_URL_MAX_LENGTH	127	Maximum length of the Avatar URL
SCE_NP_ONLINEID_MIN_LENGTH	3	Minimum length of the Online ID
SCE_NP_ONLINEID_MAX_LENGTH	16	Maximum length of the Online ID
SCE_NP_ABOUT_ME_MAX_LENGTH	63	Maximum length of the self introduction (bytes)
SCE_NP_TITLE_ID_LEN	12	Maximum length of the NP Title ID
SCE_NP_TITLE_SECRET_SIZE	128	Maximum size of the NP Title Secret data
SCE_NP_CLIENT_ID_MAX_LEN	128	Maximum length of the Client ID
SCE_NP_AUTHORIZATION_CODE_MAX_LEN	128	Maximum length of the authorization code string

Description

This constant is used to represent a data size used by the NP client libraries.

Platform Types

Platform type used in cross platforms

Definition

```
#include <np/np_common.h>
typedef SceInt32 SceNpPlatformType;
```

Value	(Number)	Description
SCE_NP_PLATFORM_TYPE_NONE	0	No platform is specified
SCE_NP_PLATFORM_TYPE_PS3	1	PlayStation®3
SCE_NP_PLATFORM_TYPE_VITA	2	PlayStation®Vita
SCE_NP_PLATFORM_TYPE_PS4	3	PlayStation®4

Description

This constant is used to represent the platform type used for titles dealing with cross platforms.

Return Codes

List of return codes returned by the NP library

Definition

Value	(Number)	Description
SCE_NP_ERROR_ALREADY_INITIALIZED	0x80550001	Attempted to initialize an already initialized library
SCE_NP_ERROR_NOT_INITIALIZED	0x80550002	Attempted to call an API when the library has not yet been initialized
SCE_NP_ERROR_INVALID_ARGUMENT	0x80550003	Argument is invalid
SCE_NP_ERROR_UNKNOWN_PLATFORM_TYPE	0x80550004	Undefined platform
SCE_NP_MANAGER_ERROR_ALREADY_INITIALIZED	0x80550501	Attempted to initialize an already initialized library
SCE_NP_MANAGER_ERROR_NOT_INITIALIZED	0x80550502	Attempted to call an API when the library has not yet been initialized
SCE_NP_MANAGER_ERROR_INVALID_ARGUMENT	0x80550503	Argument is invalid
SCE_NP_MANAGER_ERROR_OUT_OF_MEMORY	0x80550504	There is not enough memory
SCE_NP_MANAGER_ERROR_INVALID_TICKET_SIZE	0x80550505	Specified ticket size is invalid
SCE_NP_MANAGER_ERROR_INVALID_STATE	0x80550506	Attempted to call an API in an invalid state (not signed-in, sign-in processing is already running, etc.)
SCE_NP_MANAGER_ERROR_ABORTED	0x80550507	Ticket obtainment or update processing was aborted
SCE_NP_MANAGER_ERROR_VARIANT_ACCOUNT_ID	0x80550508	Ticket is invalid
SCE_NP_MANAGER_ERROR_ID_NOT_AVAIL	0x80550509	Request ID is invalid
SCE_NP_MANAGER_ERROR_SIGNOUT	0x8055050a	Called in the not signed-up state
SCE_NP_MANAGER_ERROR_NEED_CALL_NETCHECK_DIALOG	0x8055050b	Called without execution of sign-in processing by Network Check Dialog
SCE_NP_UTIL_ERROR_INVALID_ARGUMENT	0x80550601	An invalid value was specified for an argument
SCE_NP_UTIL_ERROR_INVALID_NP_ID	0x80550605	Specified NP ID is invalid
SCE_NP_UTIL_ERROR_NOT_MATCH	0x80550609	The two NP IDs that were compared were different values