NP Lookup Library Overview

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Document serial number: 000004892117

Table of Contents

1 Library Overview

Purpose and Features

The NP Lookup library is a library for obtaining information of PSN℠ users from the server. A user's Online ID can be specified to search for the user's NP ID, and when the NP ID is specified, the user's profile (including the avatar, and language preference) can be obtained.

Information of a login user, friend, and room member can be obtained from the NP library, NP Basic library, and NP matching 2 library, respectively. Use this library when you require information of a user other than those above.

Main Functions

The main functions provided by the NP Lookup library are as follows.

- A function to search for NP IDs from online IDs
- A function to retrieve profile information by specifying NP IDs.
- A function to retrieve Avatar images from retrieved Avatar URLs.

Used Resources

The NP Lookup library uses the following system resources.

Resource	Description
Footprint	Approx. 45 KiB when PRX is loaded
Work memory	(The application is not required to provide memory.)
Thread	When `sceNpLookupInit()` is called with `SCE_TRUE` set in *`usesAsync`*, a thread for asynchronous functions is generated. The above thread is stopped with `sceNpLookupTerm()`. If `SCE_FALSE` is set in *`usesAsync`*, the thread will not be generated. Thread priority and CPU affinity mask are specified from the application with the argument of `sceNpLookupInit()`.
Processor time	Can be ignored

Embedding into a Program

Include np.h in the source program. Various header files will be automatically included as well.

Load also the PRX module in the program as follows.

```
if ( sceSysmoduleLoadModule(SCE_SYSMODULE_NP_UTILITY) != SCE_OK ) {
    // Error processing
}
```

Upon building the program, link libSceNpUtility_stub.a.

Sample Programs

The following program is provided as a NP Lookup sample program for reference purposes.

sample_code/network/api_np/np_lookup/

A sample for retrieving NP ID, user profile and Avatar image.

Reference Materials

Refer to the following document for an overview of the PSN℠ functionalities.

- PSN℠ Overview

Refer to the following documents regarding the NP library, which is commonly required when using the PSN℠ functionalities.

- NP Library Overview
- NP Library Reference

Refer to the following document regarding Network Check Dialog for switching service states of the NP library.

- Network Overview

Refer to the following document regarding the system software related to the NP Lookup library.

- System Software Overview

2 Using the Library

Initialization

To use the NP Lookup library, first initialize the NP library and then initialize the NP Lookup library.

(1) Load PRX

Call `sceSysmoduleLoadModule()` with `SCE_SYSMODULE_NP_UTILITY` specified as the module ID to load the PRX.

(2) Initialize the library

Load NET, HTTP, and NP modules, and perform initialization. For details, refer to the documents of the respective libraries.

Initialize the NP Lookup library by `sceNpLookupInit()`.

```
ret = sceNpLookupInit(
                SCE_TRUE,
                SCE_KERNEL_DEFAULT_PRIORITY_USER,
                SCE_KERNEL_THREAD_CPU_AFFINITY_MASK_DEFAULT,
                NULL);
if (ret < 0) {
        // Error handling
}
```

When `SCE_TRUE` is set in the first argument *`usesAsync`*, a thread for asynchronous functions is generated internally. If `SCE_FALSE` is set in *`usesAsync`*, the thread will not be generated. If `SCE_FALSE` is set in *`usesAsync`*, the asynchronous functions `sceNpLookupNpIdAsync()`, `sceNpLookupUserProfileAsync()` and `sceNpLookupAvatarImageAsync()` cannot be used.

(3) Create an NP Lookup title context

Call `sceNpLookupCreateTitleCtx()` to create an NP lookup title context. As arguments, specify the NP Communication ID (`SceNpCommunicationId`) for identifying the application, and the NP ID (`SceNpId`) of the login user.

When NULL is passed to the NP communication ID of this function, the NP communication ID set with the `sceNpInit()` function of the NP library is used. Regarding `sceNpInit()`, refer to the "NP Library Overview" and "NP Library Reference" documents.

Note
Although data based on the NP Communication ID is not handled in current specifications, make sure to specify the correct NP Communication ID.

```
int ret, titleCtxId;
SceNpCommunicationId communicationId;
SceNpId selfNpId;
// communicationId assumed to store an appropriate value

ret = sceNpManagerGetNpId(&selfNpId);
if (ret < 0) {
        // Error handling
}
ret = sceNpLookupCreateTitleCtx(&communicationId, &selfNpId);
if (ret < 0) {
        // Error handling
```

```
}
titleCtxId = ret;
```

Communication

The NP Lookup library provides a synchronous API and an asynchronous API for obtaining user information from the server. Follow the procedure below when using the synchronous API.

(1) Create a request

Create a request ID to be used for aborting and deleting a request. A request ID must be created per request, and deleted after the request ends.

```
int ret, reqId, titleCtxId;
SceNpId npId;
SceNpOnlineId onlineId;

// titleCtxId assumed to store an appropriate value

ret = sceNpLookupCreateRequest (titleCtxId);
if (ret < 0) {
        // Error handling
}
reqId = ret;
```

(2) Request

Call the appropriate function from below according to the information you wish to obtain.

- Look up the NP ID: `sceNpLookupNpId()`
- Obtain a user's profile information and avatar image: `sceNpLookupUserProfile()`
- Obtain a user's avatar image: `sceNpLookupAvatarImage()`

These functions block other processing until the applicable information is obtained from the server.

```
// Request
strncpy(onlineId.data, "user_foo", SCE_NP_ONLINEID_MAX_LENGTH + 1);
ret = sceNpLookupNpId (
        reqId,      // Request ID
        &onlineId,  // Online ID of the target user (case-insensitive)
        &npId,      // NP ID of the target user will be stored
        NULL);
if (ret < 0) {
        // Error handling
}
....
```

(3) Destroy request

When the request ends, destroy the request ID.

```
// Destroy the request
sceNpLookupDeleteRequest (reqId);
```

Request Using an Asynchronous API

The functions below are non-blocking. They return immediately (without waiting for information to be obtained from the server) after starting the request.

- Lookup NP ID: `sceNpLookupNpIdAsync()`
- Obtain profile information: `sceNpLookupUserProfileAsync()`

- Obtain the avatar image: `sceNpLookupAvatarImageAsync()`

To confirm the termination of the request started by these functions, call `sceNpLookupWaitAsync()` or `sceNpLookupPollAsync()`. `sceNpLookupWaitAsync()` waits for the request to complete if it hasn't already done so. `sceNpLookupPollAsync()` immediately returns the value 1 if the request has not completed. If one of these APIs returns 0 and the application receives the result of the communication, the request will complete.

Requests Which Do Not Allow Parallel Execution

Asynchronous APIs are processed with an internal thread generated with `sceNpLookupInit()`. Since only one thread is generated, even if multiple asynchronous functions are called simultaneously, internally they are processed one at a time, without allowing parallel execution. If you wish to perform parallel execution, generate multiple threads by the application, and execute synchronous API in each thread. Also, as the number of requests that can exist simultaneously is 32, parallel execution of a larger number is not possible.

Termination

(1) Destroy NP Lookup title context

When the title context is no longer required, call `sceNpLookupDeleteTitleCtx()` to destroy the context.

```
int ret;
uint32_t titleCtxId;

// titleCtxId assumed to store an appropriate value

ret = sceNpLookupDeleteTitleCtx (titleCtxId);
if (ret < 0) {
        // Error handling
}
```

(2) Terminate the NP Lookup library

Call `sceNpLookupTerm()` to terminate the NP Lookup library

Note
`sceNpLookupTerm()` is not multithread safe. When `sceNpLookupTerm()` is called, the created title context and request will be deleted automatically, however, it is recommended that this function be called after the title contexts and requests are explicitly deleted from the application side.

```
// If a thread for asynchronous functions had been generated, it will be stopped.
sceNpLookupTerm();
```

(3) Other termination processing

Perform termination processing of NET, HTTP and NP modules. For details, refer to the documents of the respective libraries.

Then, unload PRX by calling `sceSysmoduleUnloadModule()` with `SCE_SYSMODULE_NP_UTILITY` specified for the module ID.

3 Server Settings

Obtain an NP Communication ID

The NP Lookup library requires an NP Communication ID to manage the title.

NP communication IDs are issued for each application by applying to the PlayStation®Vita Developer Network (https://psvita.scedev.net/).

4 Notes

Notes Regarding Server Loads and Frequency of Request Executions

To prevent excess loads on the server of PSN™, please note the following upon designing application specifications and implementations. These points should be especially noted for large-scale applications. When unsure, consult technical support through the PlayStation®Vita Developer Network website.

Avoid Re-obtaining Information

Information of a login user, friend, and room member can be obtained from the NP library, NP Basic library, and NP Matching 2 library, respectively. Use this NP Lookup library after making sure that it is necessary for obtaining the information you need.

Prevent Continual Reloads

When implementing the reload functionality, avoid a user interface where the user can repeat reloads consecutively. For example, avoid a design specification where the continual pressing of a button results in continual reloads.

Do Not Obtain Information until It Becomes Necessary

Information to be displayed by a user operation should be obtained after the user operation is carried out, wherever possible. For example, avoid obtaining information that isn't displayed until the options menu is opened or unless the user performs scrolling, beforehand.

Directly Obtain Information of the Party with whom You are Performing P2P Communication

Information of the party with whom you have P2P communication should be directly obtained wherever possible. You can obtain information of a user using the NP Lookup library before performing P2P communication with them, but even in this case, first check to make sure that this information is not one that can be obtained through the NP Matching 2 library.

Limit the Number of Requests Executed at One Time

There is no quantitative limitation on the number of requests (the number of requests that can exist at one time is up to 32 given utility specifications). However, note that if specifications are such that user operation gets blocked until multiple transactions end, the wait time may become extremely long depending on the state of the network, and smooth operation may not be possible. When enabling the setting of "★Debug Settings" - "PSN™ - NP Debug" through "Settings" of the system software, a mode producing a delay of one second after the communication request will be entered. Be sure not to disrupt the user operation even in this mode.