NP Activity Library Overview

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Table of Contents

1 Library Overview

Purpose and Features

The NP Activity library is used for posting status on the activity service of PSN℠. Using the library, status information can be set and subsequently posted to the activity server.

Status can even be posted in an environment when the network is not connected. If status is posted from a game when the device is not connected to the network, the status information will be registered and saved in a system software database. The next time the device is connected to the network, system software will automatically post the saved status to the activity server.

Before using the activity service, you must sign up to Sony Entertainment Network. If you attempt to post a status without being signed into PSN℠, your status will be discarded and will not be saved.

Main Features

The NP Activity library provides the following main features.

- Posting of status to the activity server
- Delayed posting, in which status is saved in a system software database when the device is not connected to the network and subsequently posted to the activity server once the device is connected

Used Resources

The system resources used by the NP Activity library are listed below.

Resource	Description
Footprint	Approximately 20 KiB when PRX is loaded
Work memory	(The application does not need to allocate this memory)
Threads	(No threads are created that affect the application)
Processor time	Negligible

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_ACTIVITY) != SCE_OK ) {
    // Error processing
}
```

Upon building the program, link libSceNpActivity_stub.a.

Sample Programs

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

sample_code/network/api_np/np_activity/

This sample program shows how to post activity status.

Reference Materials

For an overview of the Activity service system, refer to the following document.

- Activity System Overview

When using the action link to startup an application, the application will parse application event parameters using a function of the application utility library. For details on the application utility library, refer to the following document.

- Application Utility Overview

For an overview of PSN℠ features, refer to the following document.

- PSN℠ Overview

For information about the NP library that is commonly required when using PSN℠ features, refer to the following documents.

- NP Library Overview
- NP Library Reference

For information about system software related to the NP Activity library, refer to the following document.

- System Software Overview

2 Using the Library

Initialization

The NP Activity library must be initialized as follows before it can be used.

(1) Load the PRX

Load the NP Activity PRX module by calling sceSysmoduleLoadModule() with SCE_SYSMODULE_NP_ACTIVITY specified for the module ID.

(2) Perform initialization

Call sceNpActivityInit() to initialize the NP Activity library.

```
ret = sceNpActivityInit(
              NULL);
if (ret < 0) {
        // Error handling
}
```

Posting Status

The post function returns immediately. The actual posting of status is performed asynchronously by the system. Even though the return of the post function is immediate in general, a certain delay may occur for the inter-process communication depending on the load status of the system.

Posting status

Status can be posted after the library is initialized.

```
int ret;
SceNpActivityStatusMessage message;
SceNpActivityStatusActionUri actionUri;

// Make sure appropriate values are stored in message and actionUri

ret = sceNpActivityPostStatus (
        &message,
        &actionUri,
        NULL);
if (ret < 0) {
        // Error handling
}
```

Posting an application startup status

Use sceNpActivityPostAppStartupStatus() to post a status with an application startup action link. Carry out URI Escape processing for the application startup argument string.

```
int ret;
SceNpActivityStatusMessage message;
SceNpActivityStatusAppStartupParam appParam;

// Initialize structure
memset(&message, 0x00, sizeof(message));
memset(&appParam, 0x00, sizeof(appParam));

// Make sure appropriate values are stored in message and appParam
// URI Escape processing
// Assuming application startup argument string is stored to pAppParam
ret = sceHttpUriEscape (
        appParam.escapedParam,
        NULL,
        SCE_NP_ACTIVITY_APP_PARAM_MAX_SIZE - 1,
        pAppParam,
        NULL);
if (ret < 0) {
        // Error handling
}

ret = sceNpActivityPostAppStartupStatus (
        &message,
        &appParam,
        NULL);
if (ret < 0) {
        // Error handling
}
```

Delayed posting

The post function can be called even when the device is not connected to the network. In this case, posted status is saved by system software and later posted to the activity server at an appropriate time after the device is connected to the network.

Status that has not yet been posted is not displayed in system applications. Once status is posted to the activity server, an application can receive it from the activity server and display it.

Posting failures

It is not possible for the application to confirm that the post function completes successfully and the status was correctly posted later to the activity server by system software.

Status can be discarded in the following cases.

- The user was signed out when the status was posted.
- The user signed out while the status was being saved internally.
- The user is signed in to a subaccount, which was set to **Do not allow chat** from the master account.
- 30 or more statuses were posted while the device was not connected to the network.
- Saving of status to the database failed due to some other problem with the device.

Design your application so that it will not malfunction even if a posted status is discarded. For example, do not create an application that prevents the game from progressing because status is being used as a key for allowing the game to proceed. Likewise, do not design your application so that the player must view status information before the game can proceed.

Note
The main purpose of the activity service is to enable users to share their play history. For an overview of the service, refer to the "Activity System Overview" document.

Posting identical status

There is no problem with reposting status for a user's play history that has already been posted. However, since at most 30 statuses are saved on the activity server, you should avoid posting status at an extremely high frequency.

Termination

NP Activity library termination processing

To terminate the NP Activity library, call `sceNpActivityTerm()`.

> **Note**
> `sceNpActivityTerm()` is not multithread safe.

Miscellaneous termination processing

Unload the PRX module by calling `sceSysmoduleUnloadModule()` with `SCE_SYSMODULE_NP_ACTIVITY` specified for the module ID.

3 Server Settings

Before Using the Activity System

To post status to the activity server, you must first obtain a title ID (product number) from the PlayStation®Vita Developer Network website (https://psvita.scedev.net/).

A feature is provided which allows you to debug status postings before you obtain a title ID, or if you are using a Development Kit that is not on the network. This feature is described later.

4 Preparing the Development Kit and Testing Kit

To use the activity service, you must sign in to PSN℠. For information about signing in, refer to the "PSN℠ Overview" document.

The activity privacy setting screen will appear when you sign in so you can set your privacy settings. An account that has not had its privacy settings set cannot use the activity service.

Note
Set an appropriate value on the Development Kit and Testing Kit for the activity visibility. Note that when the visibility is set to **Anyone**, it will be possible to view status from third party's Development Kit and Testing Kit.

5 Debugging Feature

Overview

The NP Activity library provides a debugging feature which allows you to debug status postings before you obtain a title ID, or if you are using a development environment that is not on the network.

- Among the applications that are started from the APP_HOME icon or from an application that is started directly from the development host computer, the posted status will be stored in a local database on the Development Kit and will not be posted to the activity server for an application for which a title ID is not specified with the param file (param.sfo).

By following the procedure described below, you can use this feature to post status and confirm its display without communicating with the activity server.

Procedure

(1) Start the application from the APP_HOME icon or from the development host computer

Start the application from the APP_HOME icon or from a development host computer without installing the package. Be sure to start the application with no param file (param.sfo) placed under the save directory or working directory of the application. For information about APP_HOME, refer to the "System Software Overview" document.

(2) Post the status

Use the NP Activity library to post status from a game. If the application was started by the method in step (1), the status will be saved in the database on the Development Kit and will not be posted to the activity server.

When using the debug feature, `sceNpActivityPostStatus()` will terminate normally even when in signed-out state. Note that `SCE_NP_ACTIVITY_ERROR_SIGNOUT` will return if the debug feature is not used.

(3) Press the PS button

Press the PS button to display the system software LiveArea™.

(4) Display the LiveArea™ communication zone

You can display the LiveArea™ communication zone and test whether the status text is displayed correctly and that action links work correctly.

> **Note**
> When using the debugging feature, posted status will be displayed immediately in the communication zone, however, when the activity server is used, status will not be displayed until the refresh button is pressed in the communication zone.

Status that is posted using the debugging feature has the following limitations.

- All test status is deleted when PlayStation®Vita is started up.
- A maximum of 100 test statuses can be saved.
- Comment and "Like" operations cannot be performed.
- Fixed values are displayed for the online ID, avatar image and background color. The values for the user who is signed in are not displayed. The fixed values for the online ID, Avatar image and background color are ★**debug_user**, the default avatar image and the default background color, respectively.

Note

Since the debugging feature is for development use, always use the activity server to perform final tests.

6 Notes

Notes Related to Status Posting

Status destruction

The posting of status may be delayed or status might be discarded depending on the state of the device. Therefore, do not post status that would prevent the game from progressing.

Status disclosure

Status may be disclosed to other users, so be sure not to post important information for a game (such as an important keyword or character name).

Number of Posts

A maximum of 30 statuses can be saved on the activity server, therefore, avoid posting status at an extremely high frequency.

Prohibited status

For other prohibited status or non-recommended status, refer to the "Activity System Overview" document.