NP Trophy Library Overview

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Table of Contents

1 Library Overview

Purpose and Characteristics

The NP Trophy library provides APIs for the application to use the trophy system – such as, setting up trophies and unlocking trophies.

Main Features

The NP Trophy library can be utilized to install a trophy set (trophy configuration data) to the internal flash memory, and to update an already installed trophy set.

In addition, the processing to "unlock" trophies – in order that they may be given to a user according to game play results – can be carried out. Trophy icon image data and information regarding the obtained trophies can also be obtained to display a collection of trophies within the application.

Embedding into a Program

Include np.h and np_trophy_setup_dialog.h in the source program.

In addition, before calling any NP Trophy library APIs in the program, load the PRX module with the relevant libsysmodule API, as follows.

```
if ( sceSysmoduleLoadModule(SCE_SYSMODULE_NP_TROPHY) != SCE_OK ) {
   // Error handling
}
```

Upon building the program, link libSceNpTrophy_stub.a and libSceCommonDialog_stub.a.

Sample Program

A sample program that uses the NP Trophy library is provided as follows.

sample_code/network/api_np/np_trophy/

This sample exemplifies the basic usage of the NP Trophy library.

Reference Materials

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

- PSN℠ Overview

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

- NP Library Overview
- NP Library Reference

Refer to the following document regarding the overview of the trophy system and how to define trophy sets.

- Trophy System Overview

Refer to the following document regarding characteristics of the system software that are related to the NP Trophy library.

- System Software Overview

Refer to the following document regarding information common to the Common Dialog library – such as, limitations and specifications.

- Common Dialog Overview

2 Using the Library

Advance Preparations

To use the NP Trophy library, create a trophy pack file in advance and place it in the appropriate folder. For details, refer to the "Trophy System Overview" document.

Initialization

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

(1) Initialize the NP Library

Load and initialize the NP Trophy library. For details, refer to the "NP Library Overview".

(2) Load the PRX

Call `sceSysmoduleLoadModule()` with `SCE_SYSMODULE_NP_TROPHY` specified as the module ID and load the PRX module.

(3) Initialize the Library

Next, call `sceNpTrophyInit()` to initialize the NP Trophy library.

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

(4) Create a Context

Call `sceNpTrophyCreateContext()` to create a context. At this time, an NP Communication ID for identifying the application and an NP communication signature equivalent to an authentication password will be required. Use an NP Communication ID and an NP communication signature that has been issued through the PlayStation®Vita Developer Network (https://psvita.scedev.net/).

```
SceNpTrophyContext ctx = SCE_NP_TROPHY_INVALID_CONTEXT;
SceNpCommunicationId commId = {
    {'X','X','X','X','0','0','0','0','1'},
    0,
    0,
    0
};
SceNpCommunicationSignature commSign = {
    {
    0xb9,0xdd,0xe1,0x3b,0x01,0x00,0x00,0x00,
    (Middle processing omitted)
    0xd6,0xcd,0xf1,0x2f,0xfd,0x25,0x38,0xea
    }
};
ret = sceNpTrophyCreateContext(&ctx, &commId, &commSign, 0);
if (ret < 0) {
    // Error handling
}
```

Setting Up Trophy Information

Trophy information must be configured for the created context. A context to which setup processing has not been performed cannot unlock a trophy.

To perform setup processing of trophy information, follow the procedure below and use Trophy Setup Dialog.

(1) Initialize Trophy Setup Dialog

Call `sceNpTrophySetupDialogInit()` for the initialization. To the `SceNpTrophySetupDialogParam` type variable of the argument, specify the context to set up.

When Trophy Setup Dialog is initialized, the following processes are carried out.

- Install the trophy set to the internal flash memory if it isn't already.
- Update the trophies installed in the internal flash memory if an update is required.
- If a network is available, the trophy set may be synchronized.

In the following cases, the user will be notified with a dialog display.

- When the number of trophy sets installed in the internal flash memory exceeds the maximum permitted, a dialog will be displayed to confirm whether existing trophy sets can be deleted.
- When trophy setup fails, an error dialog will be automatically displayed and the game cannot be continued.
- When `SCE_NP_TROPHY_SETUP_DIALOG_OPTION_SHOW_PROCESSING_DIALOG` is specified to the `SceNpTrophySetupDialogParam` type variable *`options`* of the argument, a dialog will be displayed during the trophy setup processing.

(2) Obtain the Operation Status

Call `sceNpTrophySetupDialogGetStatus()` per frame and poll the operation status of Trophy Setup Dialog. While Trophy Setup Dialog is displayed, `SCE_COMMON_DIALOG_STATUS_RUNNING` will return as the status. Wait until this becomes `SCE_COMMON_DIALOG_STATUS_FINISHED`.

Moreover, when an operation status is during `SCE_COMMON_DIALOG_STATUS_RUNNING`, `sceCommonDialogUpdate()` must be called. For details, refer to the "Common Dialog Overview".

(3) Obtain the Setup Result

When the operation status transitions to `SCE_COMMON_DIALOG_STATUS_FINISHED`, the result of the setup processing (in other words, whether the install was successful or not) can be obtained with `sceNpTrophySetupDialogGetResult()`.

(4) Terminate Trophy Setup Dialog

Once the result of the processing is obtained, call `sceNpTrophySetupDialogTerm()` to carry out termination processing. This will release the resources allocated upon initialization.

Where to Install the Trophy Set

Most of the trophy configuration data – name and details of the trophy set, trophy group and each trophy, grade settings, hidden attribute settings, platinum link settings, trophy set's still-image icon, trophy group's still-image icon, etc. – are installed in the internal flash memory. Only the trophies' still-image icons will be stored in a memory card.

If a memory card is not inserted, trophy still-image icons will not be installed. When this happens, a default trophy still-image icon (installed in the system software) will be used in places such as the local browser of the system software where the display of a trophy still-image icon is required.

When checking whether a trophy icon is correctly displayed in the system software – such as, when defining a trophy set – install the trophy set with a memory card inserted, and perform the check.

Timing to Call the Setup Process

Normally, only perform trophy setup once after application boot. Design implementation so that APIs (such as the function to unlock a trophy) are used with the context set up upon application boot throughout the life cycle of the application.

When Failing Setup

When trophy setup processing fails, have the application wait (without doing anything) for the user to perform an application termination operation.

When the call of Trophy Setup Dialog succeeds, an error dialog will be displayed automatically when an error occurs in subsequent processing and the application cannot be continued. When **Trophy Setup Dialog Debug** in ★**Debug Settings** is used, operation when an error has occurred in the trophy setup processing can be checked.

In addition, errors that occur from `sceNpTrophyInit()`, `sceNpTrophyCreateContext()`, and `sceNpTrophySetupDialogInit()` calls will be errors that occur only during development such as argument specification mistakes. Therefore, the minimum error handling can be used in the application.

Unlocking a Trophy

(1) Initialize Handle

Prepare a variable of the `SceNpTrophyHandle` type and call `sceNpTrophyCreateHandle()` to initialize it.

```
SceNpTrophyHandle handle = SCE_NP_TROPHY_INVALID_HANDLE;
ret = sceNpTrophyCreateHandle(&handle);
if (ret < 0) {
    // Error handling
}
```

(2) Unlocking a Trophy

Call `sceNpTrophyUnlockTrophy()` to unlock a trophy. Because this function's processing takes time, call the function from a subthread. As arguments, specify the context, handle, the ID of the trophy to be given to the user, and the variable for receiving the ID of the platinum trophy.

```
SceNpTrophyId platinumId = SCE_NP_TROPHY_INVALID_TROPHY_ID;
ret = sceNpTrophyUnlockTrophy(ctx, handle, thisTrophyId, &platinumId);
if (ret < 0) {
    // Error handling
}
if (platinumId != SCE_NP_TROPHY_INVALID_TROPHY_ID) {
    // Processing to be carried out when the platinum trophy is unlocked
}
```

Unlike other trophies, the application cannot unlock the platinum trophy. It is unlocked by the system when all other trophies are unlocked, and its unlocking will be indicated by the function's return value.

> **Note**
> Refer directly to gameplay results to determine whether or not to unlock a trophy. Do not unlock a trophy based on information recorded in the save data, etc.
> Also note that trophies must not be unlocked if the game was started using another user's save data or another user's online data.

(3) Destroy Handle

Call `sceNpTrophyDestroyHandle()` to destroy the handle.

```
ret = sceNpTrophyDestroyHandle(handle);
if (ret < 0) {
    // Error handling
}
```

Aborting Processing

To abort processing of a function of the NP Trophy library that is taking time such as a function for trophy unlocking or trophy information setup, call `sceNpTrophyAbortHandle()`.

Note
When a processing is aborted with `sceNpTrophyAbortHandle()`, the used handle should be destroyed with `sceNpTrophyDestroyHandle()`. The handle can then be created again with `sceNpTrophyCreateHandle()`, and the execution of the aborted processing can be reattempted.

Termination

(1) Destroy Context

When a context is no longer necessary, call `sceNpTrophyDestroyContext()` to destroy the context.

```
int ret;
SceNpTrophyContext ctxId;

// Assuming that an appropriate value is stored in ctxId

ret = sceNpTrophyDestroyContext(ctxId);
if (ret < 0) {
    // Error handling
}
```

(2) Terminate the NP Trophy Library

Call `sceNpTrophyTerm()` to terminate the NP Trophy library.

Then, terminate the NP library.

Note
`sceNpTrophyTerm()` is not multithread safe. Although the created context and handle will be automatically deleted when this function is called, it is recommended that the application explicitly delete them before calling the function.

(3) Unload the PRX

Call `sceSysmoduleUnloadModule()` with `SCE_SYSMODULE_NP_TROPHY` specified as the module ID and unload the PRX.

Debugging Support on the System Software

The following debugging features are provided in the system software for the development of the title supporting the trophy.

Check Operations Performed When Trophy Setup Dialog Error Occurs

With this feature, it is possible to emulate the operations performed when an error occurs during the Trophy Setup Dialog setup processing. By selecting **Always Fail** from the **Trophy Setup Dialog Debug** item of ★**Debug Settings** of Settings application, the setup processing of Trophy Setup Dialog always fails and an error dialog is displayed. For details, refer to the "System Software Overview" document.

Delete All Trophy Data

This item deletes trophy data that was installed to the internal flash memory. Display the list of trophy sets from **Trophies** in the system software, and then select ★**Delete All** from the options menu.

Delete Trophy Data

This item deletes trophy data that was installed to the internal flash memory by each trophy set. Display the information screen of the trophy set you wish to delete from **Trophies** in the system software, and then select ★**Delete** from the options menu.

Lock and Unlock Individual Trophies

Individual trophies can be locked and unlocked from the options menu of each trophy information screen. This feature can be used to test the behavior of the title with certain trophies locked/unlocked.

To unlock a trophy, select the target trophy and select ★**Unlock** from the options menu. To return an unlocked trophy back to the locked state, select the target trophy and select ★**Lock** in the options menu.

It is not, however, possible to lock/unlock platinum trophies directly. Platinum trophies are automatically locked/unlocked by the system when all the trophies with the platinum link have been locked/unlocked.

It is possible to lock/unlock all the trophies included in a specific trophy set or a specific trophy group in one process. Display the list that includes the target trophies and select ★**Lock All** or ★**Unlock All** from the options menu.

Automatically Detect Trophy Set Updates and Re-Install Trophies

When the checkbox "For Development" is selected in the Trophy Pack File Utility, the option **Development Flag** will be enabled for the trophy pack files that are created.

When these trophy pack files are used, modifications to the trophy configuration data will automatically be detected and applied (re-installed). This feature can contribute to improved efficiency during development when trophy configuration data is frequently modified, since the trophy set will not need to be deleted from the system software every time that trophy configuration data is updated.

If all of the following conditions apply when Trophy Setup Dialog is called, the trophy set will be overwritten and re-installed.

- Trophy configuration data has been installed
- The trophy pack files to be installed have "Development Flag" enabled
- The version of the installed trophy set and the version of the trophy set to be installed are the same
- The trophy set has been modified

Note that if trophy configuration data is re-installed under these conditions, information of trophies earned will be lost.

"Development Flag" can be used only during development. Do not set this flag to the trophy pack files for master submission. Trophy pack files with this flag will cause an error when the Trophy Authoring Tool is used to append digital signatures.

3 Notes

Demo/Trial Versions

The trophy system cannot be used in demo/trial versions. Be sure that a demo/trial version for a trophy-supported title does not call any NP Trophy library functions.

Posting Status to the Activity System

When a user earns a trophy, the trophy record is posted as a status of the activity system. This is done automatically by the trophy system and there is no need for the application to make a status post on its own. For details, refer to the "Activity System Overview".

Handling Data Such as Save Data

Preventing Indirect Transfers of Trophies

In the trophy system, the trophies obtained by all users for all titles are displayed online for view by everyone. If trophies are easily transferable between users, there is no meaning to the trophy records and accomplishments, and the trophy system itself can be rendered obsolete.

For this reason, the trophy system binds trophy records to individual users and handles the data as non-transferable data. It is possible, however, for trophies to be transferred indirectly through save data, online save data on the network, etc.; and thus the application must take measures to prevent such transfers. In other words, the application must prevent save data with good scores from being used by other users to obtain trophies. The following issues must be taken into account.

- Create data files relating to the unlocking of trophies – such as, character data and game progress status – as protected save data files.
- When saving information that is equivalent to save data on a network storage, confirm that the data owner is the user using the Sony Entertainment Network account.

Moreover, make sure to adhere to the following prohibitions.

(a) Trophies must not be unlocked if save data of another user or online save data of another user is loaded and game progress is inherited from the data

(b) Game progress inherited from the save data of another user or online save data of another user must not be saved as the current user's save data

Regarding (a), a mechanism is provided by the system to automatically prohibit unlocking when another user's save data is being used. When online save data of another user is being used for (a), as well as (b), must be handled by the application.

System Measures

The NP Trophy library provides an automatic feature that prevents trophies from being unlocked when save data contains no right to obtain trophies.

Save data records "information on the rights to obtain trophies" indicating the user who first created the save data. The information on the rights to obtain trophies does not get modified when save data is overwritten by another user.

This information on the rights to obtain trophies is used to determine when trophies should not be unlocked. The conditions that trigger this automatic ban on unlocking are as follows.

- When save data mounted at application start-up does not contain information on the current user's rights to obtain trophies

Once this condition is true, the unlock API of the NP Trophy library, `sceNpTrophyUnlockTrophy()`, will always return an error.