

NP Friend List Dialog Overview

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

Table of Contents

1 Library Overview.....	3
Scope of This Document.....	3
Purpose and Features.....	3
Main Features	3
Embedding into a Program	3
Sample Program	3
Reference Material	3
2 Usage Procedure	4
Basic Usage Procedure	4
NP Friend List Dialog Library Call Procedure	5
3 Reference Information.....	6
Conditions for Use.....	6
Libraries to be Loaded Beforehand and Initialization.....	6
Info Bar, Background Color and Dimmer	6
4 Notes	7
Limitations	7

1 Library Overview

Scope of This Document

This document describes the NP Friend List Dialog library that is used to display the list of shared play history managed by the system and to obtain the NP ID of the player selected by the user. The following features are realized by the NP Friend List Dialog library.

- Feature to select the player by displaying the shared play history

Purpose and Features

The NP Friend List Dialog library is the library that enables the users to select the desired players from among the shared play history list managed by the system

The NP Friend List Dialog library is one of the Common Dialog library features, and encapsulates the GUI display and user operation handling. The main utilization flow consists in first specifying and calling the selection mode, monitoring the closing of the dialog through polling, and last obtaining the call result.

Main Features

The following are the main features provided by the NP Friend List Dialog library.

- Feature to select the player by displaying the shared play history

Note

Previously, the NP Friend List Dialog library also provided a feature for displaying friends lists and selecting players. In accordance with the limit on the number of friends in a friends list on PSNSM list being increased to 2000, this feature has been moved to the NP Friend List2 Dialog library and removed from the NP Friend List Dialog library. For details, refer to the technote "Removal of APIs That Handle Information for a Maximum of 100 Friends" (<https://psvita.scedev.net/technotes/view/423>).

In addition, refer to the following document for details on the NP Friend List2 Dialog library.

- NP Friend List2 Dialog Overview

Embedding into a Program

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

The PRX module need not be loaded.

Upon building the program, link libSceCommonDialog_stub.a.

Sample Program

A sample program that uses the NP Friend List Dialog library is provided as follows.

sample_code/system/api_np_friendlist_dialog/

This sample calls NP Friend List Dialog and displays NP Profile Dialog of the selected player.

Reference Material

For common limitations, specifications, etc., of the Common Dialog library, refer to the following document.

- Common Dialog Overview

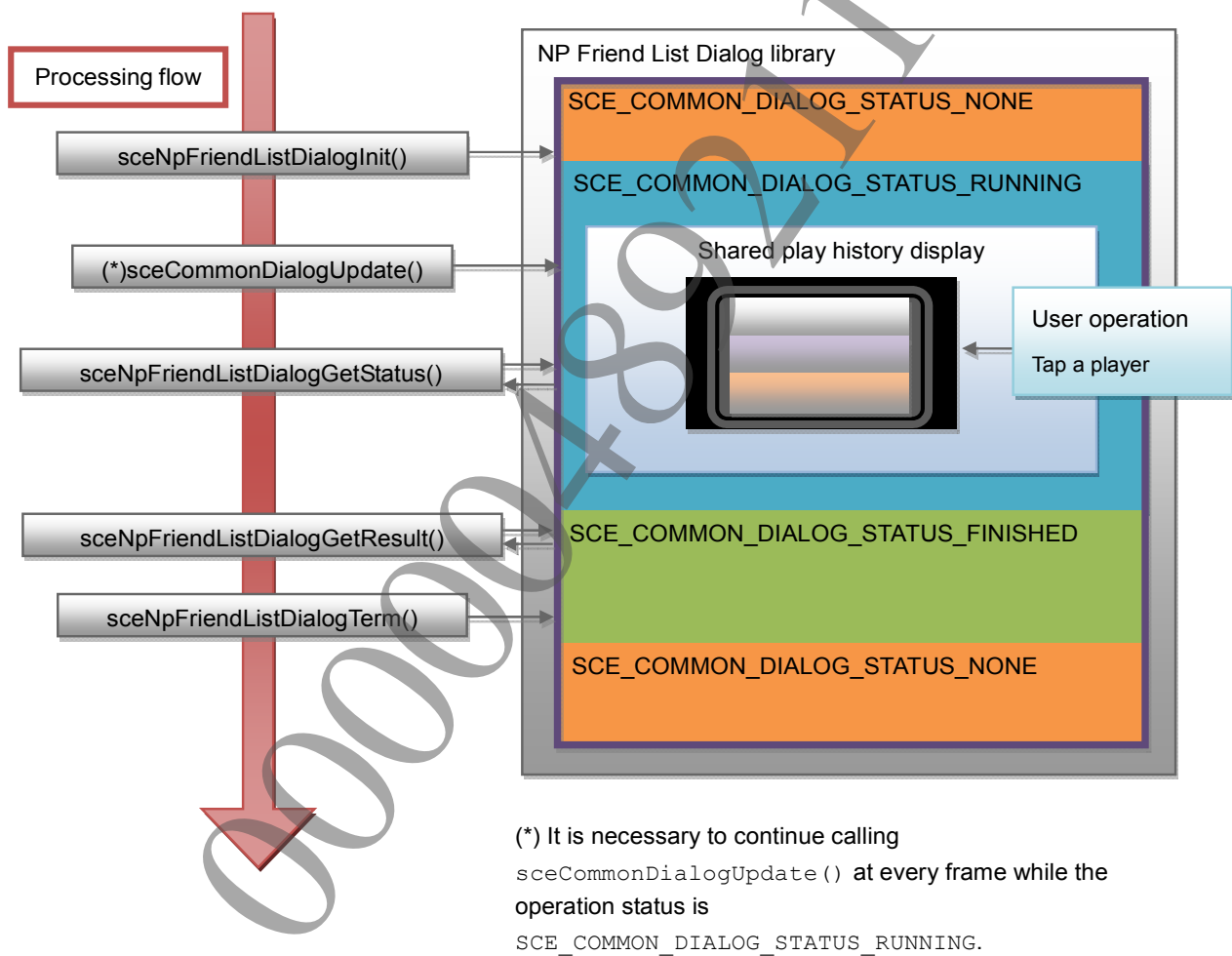
2 Usage Procedure

Basic Usage Procedure

The basic procedure to call the NP Friend List Dialog library is described below. The processing flow is outlined as follows.

- (1) Set the parameters to the variables of the `SceNpFriendListDialogParam` type
- (2) Call a feature
- (3) Wait for the response from the dialog
- (4) Obtain the call result
- (5) Terminating processing

Figure 1 Basic Processing Procedure



NP Friend List Dialog Library Call Procedure

First, prepare the `SceNpFriendListDialogParam` type variable and following initialization with `sceNpFriendListDialogParamInit()`, be sure to set the operation mode (*mode*).

(1) Calling the feature

Call an NP Friend List Dialog feature with `sceNpFriendListDialogInit()`. Specify the `SceNpFriendListDialogParam` type variable set beforehand as the argument.

(2) Waiting for the response from the dialog

Call `sceNpFriendListDialogGetStatus()` to poll the operation status of NP Friend List Dialog at each frame.

Note

It is necessary to call `sceCommonDialogUpdate()` at every frame while the operation status is `SCE_COMMON_DIALOG_STATUS_RUNNING`. For details, refer to the "Common Dialog Overview" document.

(3) Obtain the call result

When the operation status changes to `SCE_COMMON_DIALOG_STATUS_FINISHED`, the result can be obtained with `sceNpFriendListDialogGetResult()`. The result that can be obtained includes the selection result at the time of the call (whether the user made a selection, whether it was canceled), and the `SceNpId` type ID of the selected player. The ID of the selected player is set in *npId* of `SceNpFriendListDialogResult`.

(4) Terminating the processing

When the operation status becomes `SCE_COMMON_DIALOG_STATUS_FINISHED`, call `sceNpFriendListDialogTerm()` to terminate the processing. As a result, the resources allocated upon calling the NP Friend List Dialog feature are released, and the operation status becomes `SCE_COMMON_DIALOG_STATUS_NONE`.

Aborting the Processing

When aborting the display of NP Friend List Dialog from the application side (when terminating the application, for example), call `sceNpFriendListDialogAbort()`. Processing is started to terminate the display and the operation status will change to `SCE_COMMON_DIALOG_STATUS_FINISHED` after the completion of the termination processing. In this case, too, the call result will be obtained with `sceNpFriendListDialogGetResult()`. `SCE_COMMON_DIALOG_RESULT_ABORTED` is returned as the obtained result.

Main APIs and Structures Used for Basic Processing

API	Description
<code>SceNpFriendListDialogParam</code>	Parameter structure such as mode setting
<code>sceNpFriendListDialogParamInit()</code>	Initializes parameter structure
<code>sceNpFriendListDialogInit()</code>	Calls feature
<code>sceNpFriendListDialogGetStatus()</code>	Gets operation status
<code>SceNpFriendListDialogResult</code>	Structure storing call result
<code>sceNpFriendListDialogGetResult()</code>	Gets call result
<code>sceNpFriendListDialogTerm()</code>	Terminates calling of feature
<code>sceNpFriendListDialogAbort()</code>	Aborts calling of feature

3 Reference Information

Conditions for Use

For the operation of the NP Friend List Dialog library, it is required to be in the signed-in state by using Network Check Dialog beforehand. Also, the network connection environment is necessary.

For details, refer to the "Network Overview" document.

When starting up NP Friend List Dialog, a larger remaining amount of each memory pool for libssl and libhttp must be secured than the prescribed amount. The size that is actually required will be provided through the following macros of sdk/target/include/np_friendlist_dialog.h.

- SCE_NP_FRIENDLIST_DIALOG_LEAST_HTTP_POOL_SIZE
- SCE_NP_FRIENDLIST_DIALOG_LEAST_SSL_POOL_SIZE

It is not a problem if the remaining amount of each memory pool for libssl and libhttp is different before and after the operation of NP Friend List Dialog.

Libraries to be Loaded Beforehand and Initialization

To use the NP Friend List Dialog library, libraries that are loaded by calling the following APIs must be loaded beforehand.

- sceSysmoduleLoadModule(SCE_SYSMODULE_HTTPS)
- sceSysmoduleLoadModule(SCE_SYSMODULE_NP_UTILITY)

The following initialization functions must be called.

- sceSslInit()
- sceHttpInit()
- sceNpInit()

Info Bar, Background Color and Dimmer

In the NP Friend List Dialog library, info bar is displayed to inform the user that the communication is in progress. Therefore, only NULL can be set to *infoBarParam*, which is an argument relating to info bar.

Also, the background color and dimmer can be specified using the common parameters for Common Dialogs. If no value is specified, the system default value (recommended) will be applied. For details, refer to the "Common Dialog Reference" document.

SCE CONFIDENTIAL

4 Notes

Limitations

Common Dialog limitations apply.

000004892117