

# InvitationDialog Library Overview

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

## Table of Contents

---

<b>1 Library Overview.....</b>	<b>3</b>
Purpose and Characteristics .....	3
Main Features .....	3
Embedding into a Program .....	3
Sample Program .....	4
Reference Materials .....	4
<b>2 Using the Dialog .....</b>	<b>5</b>
Preparation .....	5
Basic Usage Procedure .....	6
Procedure for Obtaining Invitation Data .....	8
<b>3 Reference Information.....</b>	<b>11</b>
Starting an Application from the Messages Application .....	11
Display of Game Icons and Title Names for Titles Under Development.....	11

# 1 Library Overview

## Purpose and Characteristics

The InvitationDialog library provides features for sending/receiving invitations to sessions.

In addition to providing features to send/receive invitations to sessions between differing platforms of PlayStation®Vita and PlayStation®4, this library provides a feature to receive game boot messages (messages with invitation data attachments) sent from the NP Message feature.

The InvitationDialog library is one of the Common Dialog library features. It conceals GUI display and the handling of user operations. The main usage flow entails calling the dialog with display contents specified, monitoring the closing of the dialog by polling, and obtaining the call result.

### Note

The "game boot message (message with invitation data attachment)" of the NP Message feature is equivalent to the "invitation data" of this library and Web APIs.

### Note

Throughout this document, the feature to send/receive messages provided by the NP Message library and the NP Message dialog will be referred to as the "NP Message feature".

## Main Features

The main features provided by the InvitationDialog library are as follows.

- Sending/receiving invitations
- Receiving game boot messages (messages with invitation data attachments) sent from the NP Message feature
- Displaying a list of received invitations and each invitation data
- Obtaining the session ID of the sender
- Joining a session
- Setting a flag for used invitation data

### Note

Of the above features, the application must use Web APIs to use features other than "receiving game boot messages (messages with invitation data attachments) sent from the NP Message feature". For details on Web APIs, refer to the "Session/Invitation Overview", "Session/Invitation Web APIs Reference", "NpWebApi Library Overview" and "NpWebApi Library Reference" documents.

## Embedding into a Program

Include invitation\_dialog.h in the source program.

The PRX module need not be loaded.

Upon building the program, link libSceCommonDialog\_stub.a and libSceNpWebApiCommonDialog\_stub.a.

The NpWebApi library must be used for the application to use Web APIs. Regarding how to embed the NpWebApi library into a program, refer to the "NpWebApi Library Overview" document.

## Sample Program

Sample program is as follows.

### **sample\_code/system/api\_invitation\_dialog/**

This sample uses InvitationDialog library to send invitations and display a list of received invitations.

## Reference Materials

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

- Common Dialog Overview

For a general description of PSN<sup>SM</sup>, refer to the following document.

- PSN<sup>SM</sup> Overview

For common NP libraries required to use PSN<sup>SM</sup> features, refer to the following documents.

- NP Library Overview
- NP Library Reference

Regarding the Network Check dialog - which switches the service state of the NP library - refer to the following document.

- Network Overview

For the system software related to the InvitationDialog library, refer to the following document.

- System Software Overview

For an overview of the Session/Invitation feature and details of Web APIs, refer to the following documents.

- Session/Invitation Overview
- Session/Invitation Web APIs Reference

For the method to use Web APIs directly from within the game, refer to the following documents.

- NpWebApi Library Overview
- NpWebApi Library Reference

Regarding features of NP Message, refer to the following documents.

- NP Message Overview
- NP Message Reference

## 2 Using the Dialog

### Preparation

#### Obtain and Place the nptitle.dat File

The nptitle.dat file is required for the application to use the NpWebApi library and the InvitationDialog library. For details on the nptitle.dat file, refer to the "NpWebApi Library Overview" document.

#### Load and Initialize Dependent Libraries

To use the InvitationDialog library, the following APIs must be called in advance to have the libraries, to which the InvitationDialog library is dependent, loaded and initialized.

- sceSysmoduleLoadModule (SCE\_SYSMODULE\_NET);
- sceSysmoduleLoadModule (SCE\_SYSMODULE\_HTTPS);
- sceSysmoduleLoadModule (SCE\_SYSMODULE\_NP);
- sceSysmoduleLoadModule (SCE\_SYSMODULE\_NP\_BASIC);
- sceSysmoduleLoadModule (SCE\_SYSMODULE\_NP\_UTILITY);
- sceNetInit();
- sceNetCtlInit();
- sceSslInit();
- sceHttpInit();
- sceNpInit();
- sceNpBasicInit();
- sceNpLookupInit();

Moreover, to use in combination with the NpWebApi library, the NpWebApi library must be loaded and initialized. For details, refer to the "NpWebApi Library Overview" and "NpWebApi Library Reference" documents.

#### Set the Service State of the NP Library

To use the InvitationDialog library, it is necessary to have the service state of the NP library set to the online state by starting the Network Check dialog in the PSN<sup>SM</sup> online mode. For details, refer to the "Network Overview" document.

#### Allocate Memory Pools of libssl and libhttp

When calling the invitation dialog, the available spaces in the memory pools of libssl and libhttp must at least be the required sizes.

These actually required sizes are indicated by the following macros of sdk/target/include/invitation\_dialog.h.

- SCE\_INVITATION\_DIALOG\_LEAST\_SSL\_POOL\_SIZE  
Size required for the memory pool of libssl
- SCE\_INVITATION\_DIALOG\_LEAST\_HTTP\_POOL\_SIZE  
Size required for the memory pool of libhttp

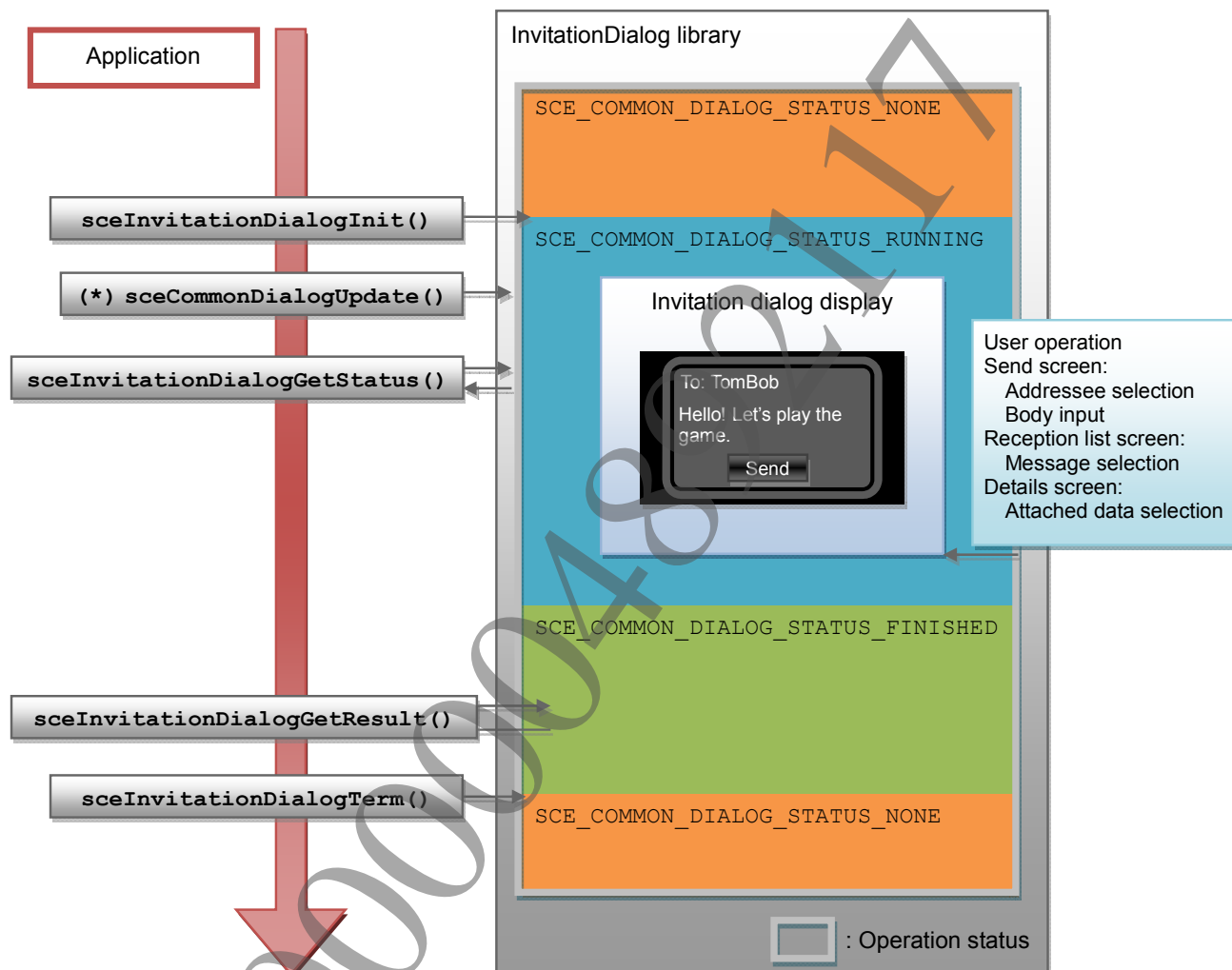
There are cases when the available memory pool spaces of libssl and libhttp differ before and after the invitation dialog operation; this is not a problem.

## Basic Usage Procedure

The basic procedure to use the InvitationDialog library is described below. An outline of the processing flow is as follows.

- (1) Set call parameters
- (2) Call the invitation dialog
- (3) Wait for the invitation dialog to be closed by polling and obtain the operation status
- (4) Obtain the call result
- (5) Terminate the invitation dialog

**Figure 1 Basic Usage Procedure**



(\*) It is necessary to continue calling `sceCommonDialogUpdate()` at every frame while the operation status is `SCE_COMMON_DIALOG_STATUS_RUNNING`.

### (1) Set parameters

Prepare the `SceInvitationDialogParam` structure, initialize it with `sceInvitationDialogParamInit()`, set the operation mode (send mode or receive mode) of the invitation dialog to the `mode` member, and set necessary parameters based on the operation mode.

Next, prepare the `SceInvitationDialogDataParam` union; fill it with 0's, and then set the necessary parameters.

When calling the invitation dialog in the send mode, set the *SendInfo* member with information required for sending invitations, such as, a list of addressees and session IDs.

When calling the invitation dialog in the receive mode, set options as necessary to the *RecvInfo* member.

## (2) Call the invitation dialog

Call `sceInvitationDialogInit()` with the `SceInvitationDialogParam` structure set in (1) as an argument. This will cause the invitation dialog to be displayed, and user operation can be received.

When the invitation dialog is called in the send mode, a send dialog will be displayed showing the list of addressees, title name, name of platforms that can use the session, session name, session's status string, session image, etc.

When the invitation dialog is called in the receive mode, a list of received invitations will be displayed. When the user selects an invitation from the list, a transition will be made to the screen displaying invitation details. This screen shows the title name, name of platforms that can use the session, session name, session's status string, session image, number of users in the session, list of users in the session, maximum number of users that can join the session, elapsed time since session creation, etc.

## (3) Obtain the operation status

Call `sceInvitationDialogGetStatus()` at every frame and obtain the operation status of the invitation dialog by polling. While the invitation dialog is displayed, `SCE_COMMON_DIALOG_STATUS_RUNNING` will return as the operation status; wait until the operation status becomes `SCE_COMMON_DIALOG_STATUS_FINISHED`.

### 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.

## (4) Obtain the call result

When the invitation dialog is closed, the operation status will transition to `SCE_COMMON_DIALOG_STATUS_FINISHED`. Call `sceInvitationDialogGetResult()` and obtain the call result of the invitation dialog.

## (5) Terminate the invitation dialog

After the call result is obtained, call `sceInvitationDialogTerm()` to terminate the invitation dialog. This will free resources allocated upon initialization and the operation status will transition to `SCE_COMMON_DIALOG_STATUS_NONE`.

## Abort Processing

To perform an emergency abort of the invitation dialog display from the application during times, such as, application termination, call `sceInvitationDialogAbort()`. Display will end immediately and the operation status will transition to `SCE_COMMON_DIALOG_STATUS_FINISHED`. It will still be possible to obtain the call result with `sceInvitationDialogGetResult()`; the call result will be `SCE_COMMON_DIALOG_RESULT_ABORTED`.

**Main APIs Used for Basic Processing**

APIs and Structures	Description
SceInvitationDialogParam	Structure storing invitation dialog call parameters
sceInvitationDialogParamInit()	Function that initializes the parameter structure
SceInvitationDialogDataParam	Structure storing send/receive invitation data parameters
sceInvitationDialogInit()	Function that calls the invitation dialog
sceInvitationDialogGetStatus()	Function that obtains the operation status
sceInvitationDialogGetResult()	Function that obtains the call result of the invitation dialog
sceInvitationDialogTerm()	Function that terminates the invitation dialog
sceInvitationDialogAbort()	Function that aborts the invitation dialog

**Procedure for Obtaining Invitation Data**

When an invitation is selected on the invitation dialog that was opened in the receiving mode, one of the following two events will be sent from the application manager.

Event	Sender
SCE_APPUTIL_APEVENT_TYPE_SESSION_INVITATION	Session/Invitation Web APIs or the InvitationDialog library
SCE_APPUTIL_APEVENT_TYPE_NP_INVITE_MESSAGE	NP Message features

When SCE\_APPUTIL\_APEVENT\_TYPE\_SESSION\_INVITATION is notified, obtain invitation data using the Session/Invitation Web APIs. An overview of the procedure for doing so is summarized in Figure 2.

When SCE\_APPUTIL\_APEVENT\_TYPE\_NP\_INVITE\_MESSAGE is notified, obtain the game boot message (message with invitation data attachment) using the NP Message library. An overview of the procedure for doing so is summarized in Figure 3.

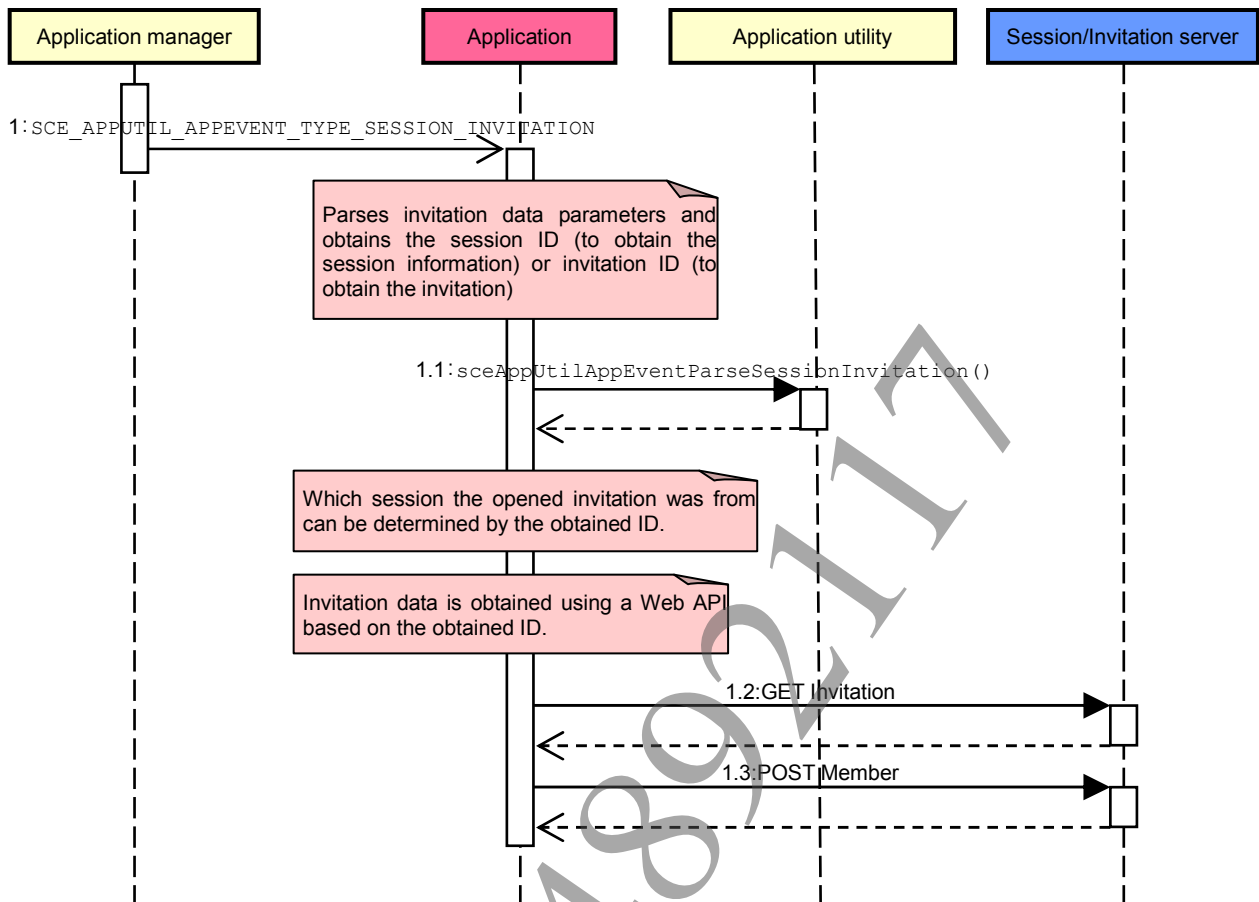
Note that an application that has never sent/received a message with invitation data attachment using the NP Message feature will not receive messages with invitation data attachments from the NP Message feature. It is possible for an application, that was using the NP Message feature before to send/receive messages with invitation data attachments before, to receive messages with invitation data attachments sent from the NP Message feature when it replaces the NP Message Dialog library with the InvitationDialog library (or when it replaces the NP Message library with Session/Invitation Web APIs).

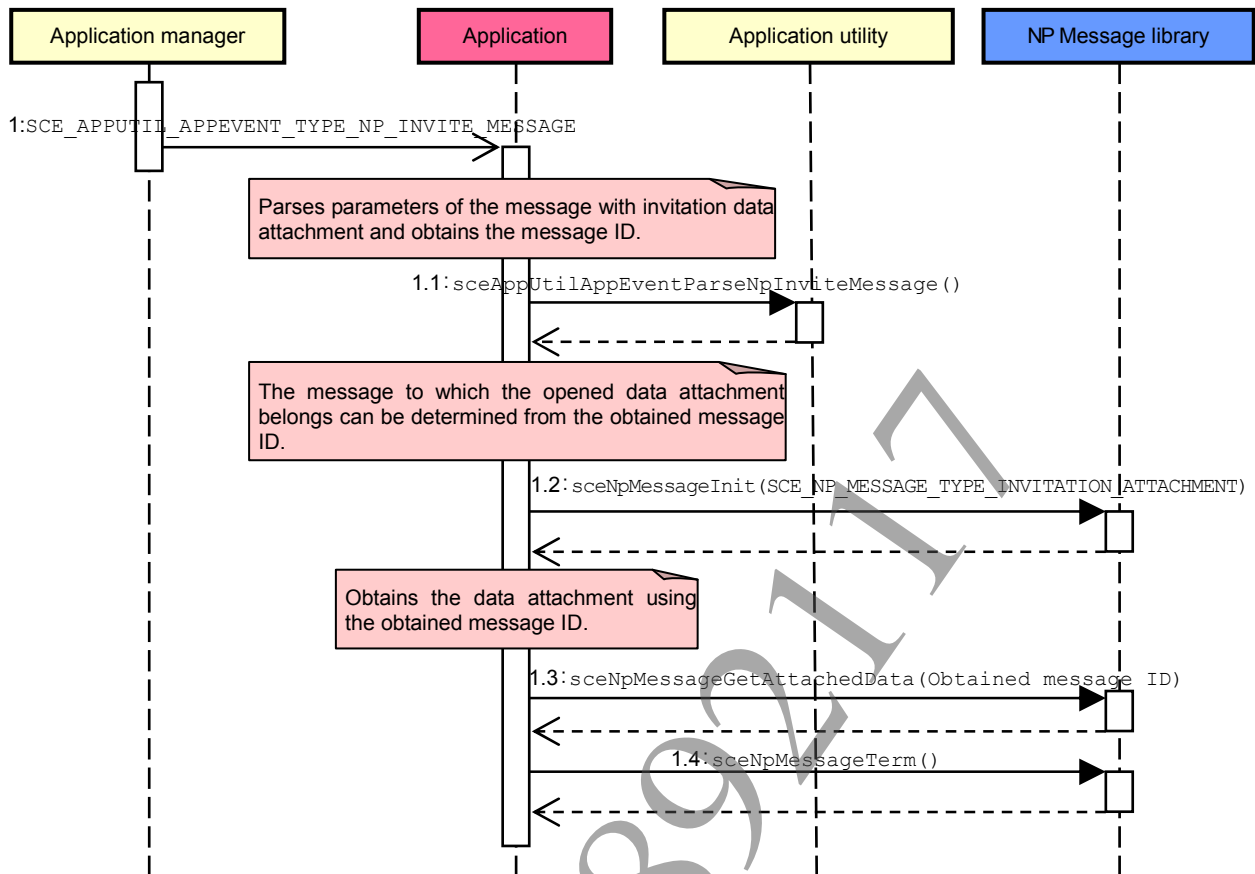
Moreover, to receive messages with invitation data attachment sent from the NP Message feature using the InvitationDialog library (or a Session/Invitation Web API), an application must continue using the same NP Communication ID as before.

Regarding the NP Message feature, refer to the "NP Message Overview" and "NP Message Reference" documents.



**Figure 2 Procedure to Obtain Invitation Data Sent from a Session/Invitation Web API or InvitationDialog Library**



**Figure 3 Procedure to Obtain Message with Invitation Data Attachment Sent from the NP Message Feature**

---

## 3 Reference Information

---

### Starting an Application from the Messages Application

When the user opens invitation data using the Messages application, the applicable application will start up.

The `SCE_APPUTIL_APEVENT_TYPE_SESSION_INVITATION` event or `SCE_APPUTIL_APEVENT_TYPE_NP_INVITE_MESSAGE` event will be sent from the application manager to the application immediately after application startup. Invitation data can subsequently be obtained in the same manner as described in the "Procedure for Obtaining Invitation Data" section.

### Display of Game Icons and Title Names for Titles Under Development

In the invitation lists, invitation screens, session lists, session screens, etc. that are displayed by the system software and invitation dialog, game icons and game title names for titles under development may not be displayed properly.

Game icons and game title names displayed by the invitation dialog, etc. can be obtained from the server of PSN<sup>SM</sup>, but application information for titles under development is not registered on the server of PSN<sup>SM</sup>.

Application information is uploaded to the server of PSN<sup>SM</sup> by SCE after submitting the application master submission and passing internal SCE QA. Therefore, game icons and title names will be properly displayed after application information is uploaded to the server of PSN<sup>SM</sup>.