

GameCustomDataDialog Library Overview

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

Table of Contents

1 Library Overview.....	3
Purpose and Characteristics	3
Main Features	3
Embedding into a Program	3
Sample Program	4
Reference Materials	4
2 Using the Dialog	5
Preparation	5
Basic Usage Procedure	6
Procedure for Obtaining Game Custom Data	8
3 Reference Information.....	11
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 GameCustomDataDialog library provides features for sending/receiving game custom data.

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

The GameCustomDataDialog 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 custom data attachment)" of the NP Message feature is equivalent to the "game custom 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 GameCustomDataDialog library are as follows.

- Sending/receiving game custom data
- Sending/receiving game custom data with URL attachments
- Displaying a list of game custom data and each game custom data
- Setting a flag for used game custom data
- Receiving game boot messages (messages with custom data attachments) sent from the NP Message feature

Note

Web APIs are required for the application to receive game custom data and to set a flag for used game custom data. For details on Web APIs, refer to the "Game Custom Data Overview", "Game Custom Data Web APIs Reference", "NpWebApi Library Overview" and "NpWebApi Library Reference" documents.

Embedding into a Program

Include game_custom_data_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_game_custom_data_dialog/

This sample uses the GameCustomDataDialog library to send game custom data and display a list of received game custom data.

Reference Materials

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

- Common Dialog Overview

For a general description of PSNSM, refer to the following document.

- PSNSM Overview

For common NP libraries required to use PSNSM 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 GameCustomDataDialog library, refer to the following document.

- System Software Overview

For an overview of the game custom data feature and details of Web APIs, refer to the following documents.

- Game Custom Data Overview
- Game Custom Data 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 GameCustomDataDialog library. For details on the nptitle.dat file, refer to the "NpWebApi Library Overview" document.

Load and Initialize Dependent Libraries

To use the GameCustomDataDialog library, the following APIs must be called in advance to have the libraries, to which the GameCustomDataDialog 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 GameCustomDataDialog 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 PSNSM online mode. For details, refer to the "Network Overview" document.

Allocate Memory Pools of libssl and libhttp

When calling the game custom data 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/game_custom_data_dialog.h.

- SCE_GAME_CUSTOM_DATA_DIALOG_LEAST_SSL_POOL_SIZE
Size required for the memory pool of libssl
- SCE_GAME_CUSTOM_DATA_DIALOG_LEAST_HTTP_POOL_SIZE
Size required for the memory pool of libhttp

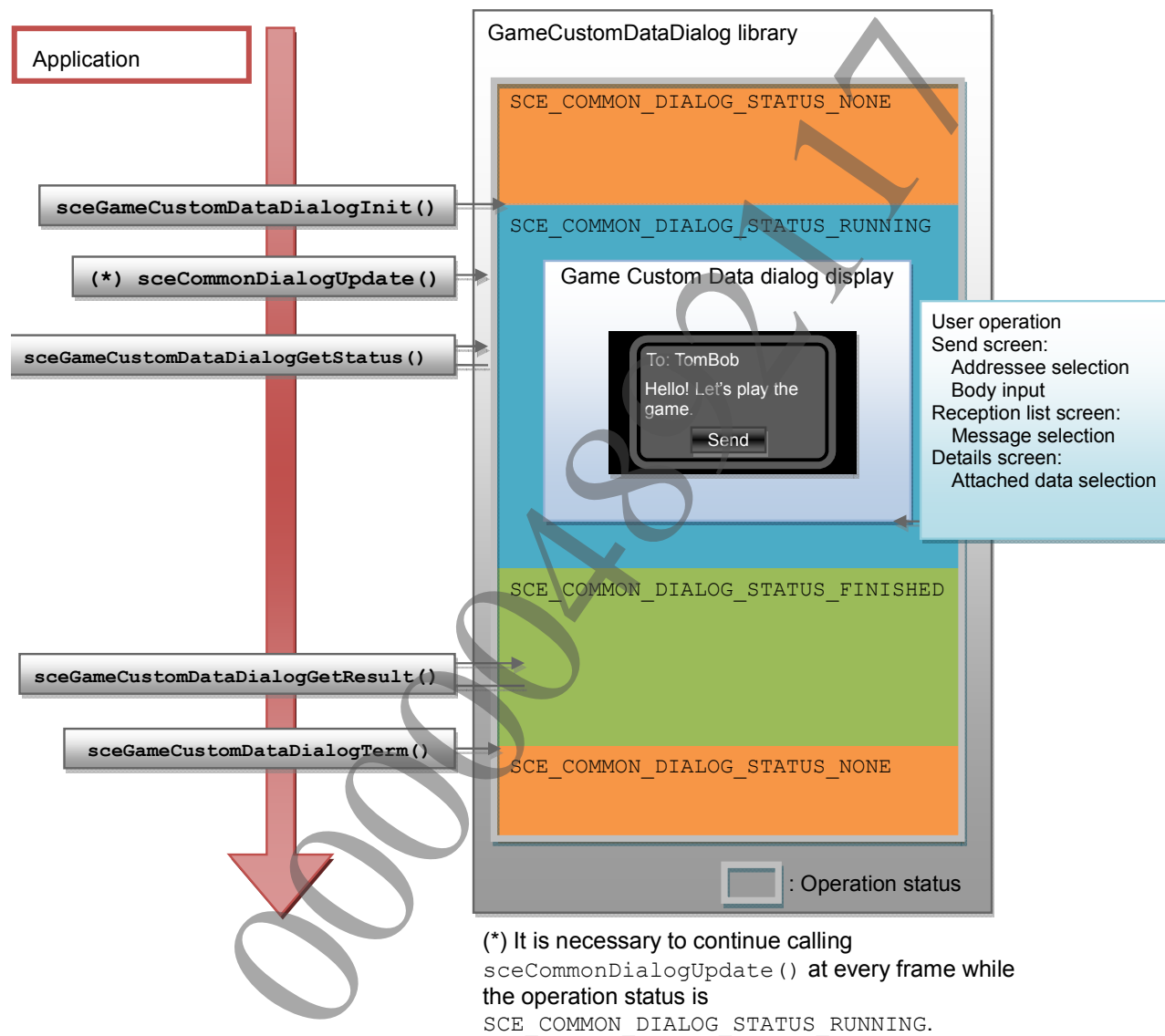
There are cases when the available memory pool spaces of libssl and libhttp differ before and after the game custom data dialog operation; this is not a problem.

Basic Usage Procedure

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

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

Figure 1 Basic Usage Procedure



(1) Set parameters

Prepare the `SceGameCustomDataDialogParam` structure, initialize it with `sceGameCustomDataDialogParamInit()`, set the operation mode of the game custom data dialog to the `mode` member, and set necessary parameters based on the operation mode.

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

When calling the game custom data dialog in the send mode, set the *SendInfo* member (*SendUrlInfo* member for sending with a URL attachment) with information required for sending game custom data, such as, a list of addressees and the message string.

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

(2) Call the game custom data dialog

Call `sceGameCustomDataDialogInit()` with the `SceGameCustomDataDialogParam` structure set in (1) as an argument. This will cause the game custom data dialog to be displayed, and user operation can be received.

(3) Obtain the operation status

Call `sceGameCustomDataDialogGetStatus()` at every frame and obtain the operation status of the game custom data dialog by polling. While the game custom data 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 game custom data dialog is closed, the operation status will transition to `SCE_COMMON_DIALOG_STATUS_FINISHED`. Call `sceGameCustomDataDialogGetResult()` and obtain the call result of the game custom data dialog.

(5) Terminate the game custom data dialog

After the call result is obtained, call `sceGameCustomDataDialogTerm()` to terminate the game custom data 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 game custom data dialog display from the application during times, such as, application termination, call `sceGameCustomDataDialogAbort()`. 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 `sceGameCustomDataDialogGetResult()`; the call result will be `SCE_COMMON_DIALOG_RESULT_ABORTED`.

Main APIs Used for Basic Processing

APIs and Structures	Description
<code>SceGameCustomDataDialogParam</code>	Structure storing game custom data dialog call parameters
<code>sceGameCustomDataDialogParamInit()</code>	Function that initializes the parameter structure
<code>SceGameCustomDataDialogDataParam</code>	Structure storing data parameters of send/receive game custom data
<code>sceGameCustomDataDialogInit()</code>	Function that calls the game custom data dialog
<code>sceGameCustomDataDialogGetStatus()</code>	Function that obtains the operation status
<code>sceGameCustomDataDialogGetResult()</code>	Function that obtains the operation status
<code>sceGameCustomDataDialogTerm()</code>	Function that terminates the game custom data dialog
<code>sceGameCustomDataDialogAbort()</code>	Function that aborts the game custom data dialog

Procedure for Obtaining Game Custom Data

When a game custom data is selected on the game custom data dialog that was opened in the receive mode, one of the following two events will be sent from the application manager.

Event	Sender
SCE_APPUTIL_APPEVENT_TYPE_GAME_CUSTOM_DATA	Game Custom Data Web APIs or the GameCustomDataDialog library
SCE_APPUTIL_APPEVENT_TYPE_NP_APP_DATA_MESSAGE	NP Message features

When SCE_APPUTIL_APPEVENT_TYPE_GAME_CUSTOM_DATA is notified, obtain game custom data using Game Custom Data Web APIs. An overview of the procedure for doing so is summarized in Figure 2.

When SCE_APPUTIL_APPEVENT_TYPE_NP_APP_DATA_MESSAGE is notified, obtain the game boot message (message with custom 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 custom data attachment using the NP Message feature will not receive messages with custom data attachments from the NP Message feature (meaning the SCE_APPUTIL_APPEVENT_TYPE_NP_APP_DATA_MESSAGE event will not be received.). It is possible for an application, that was using the NP Message feature before to send/receive messages with custom data attachments before, to receive messages with custom data attachments sent from the NP Message feature when it replaces the NP Message Dialog library with the GameCustomDataDialog library (or when it replaces the NP Message library with Game Custom Data Web APIs).

Moreover, to receive messages with game custom data attachment sent from the NP Message feature using the GameCustomDataDialog library (or a Game Custom Data 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 Game Custom Data Sent from a Game Custom Data Web API or GameCustomDataDialog Library

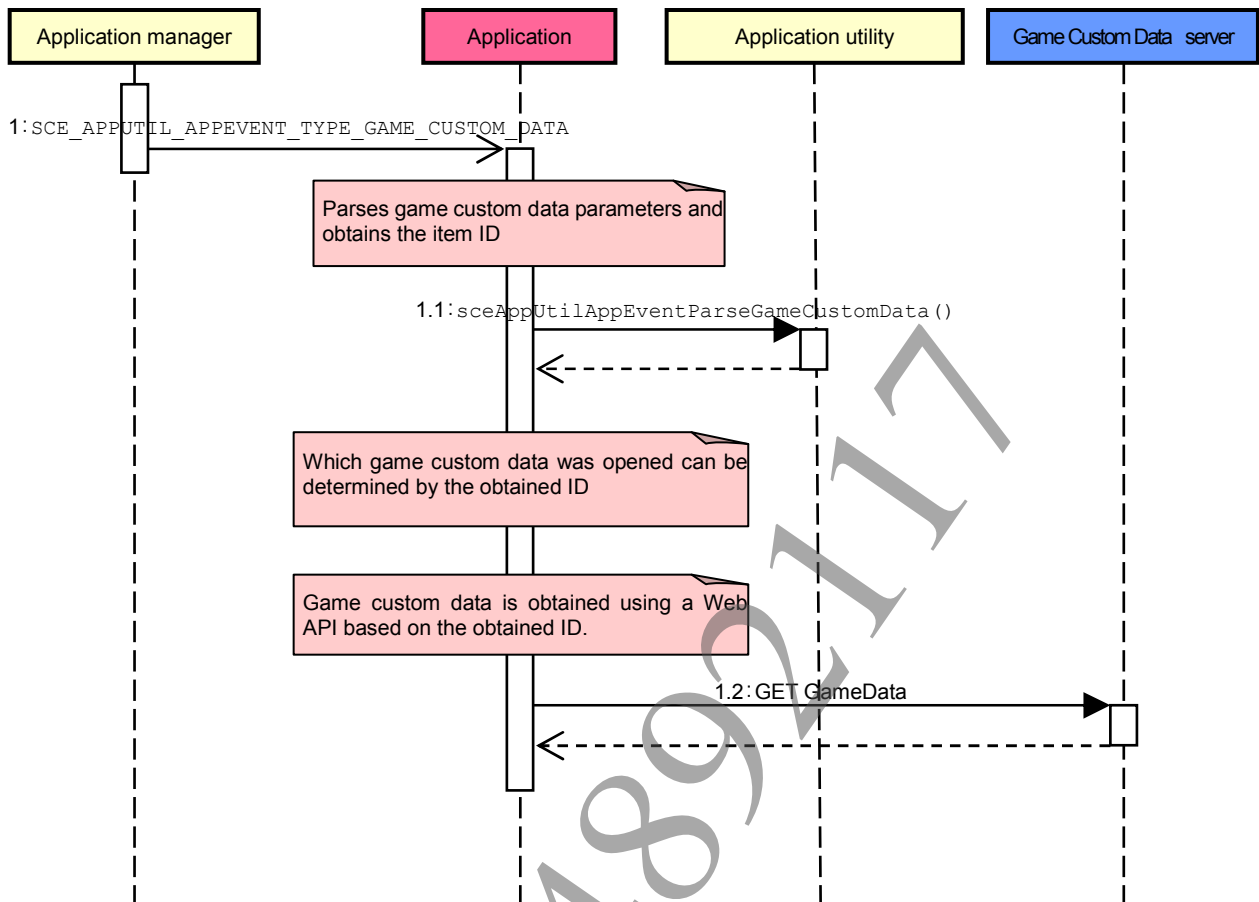
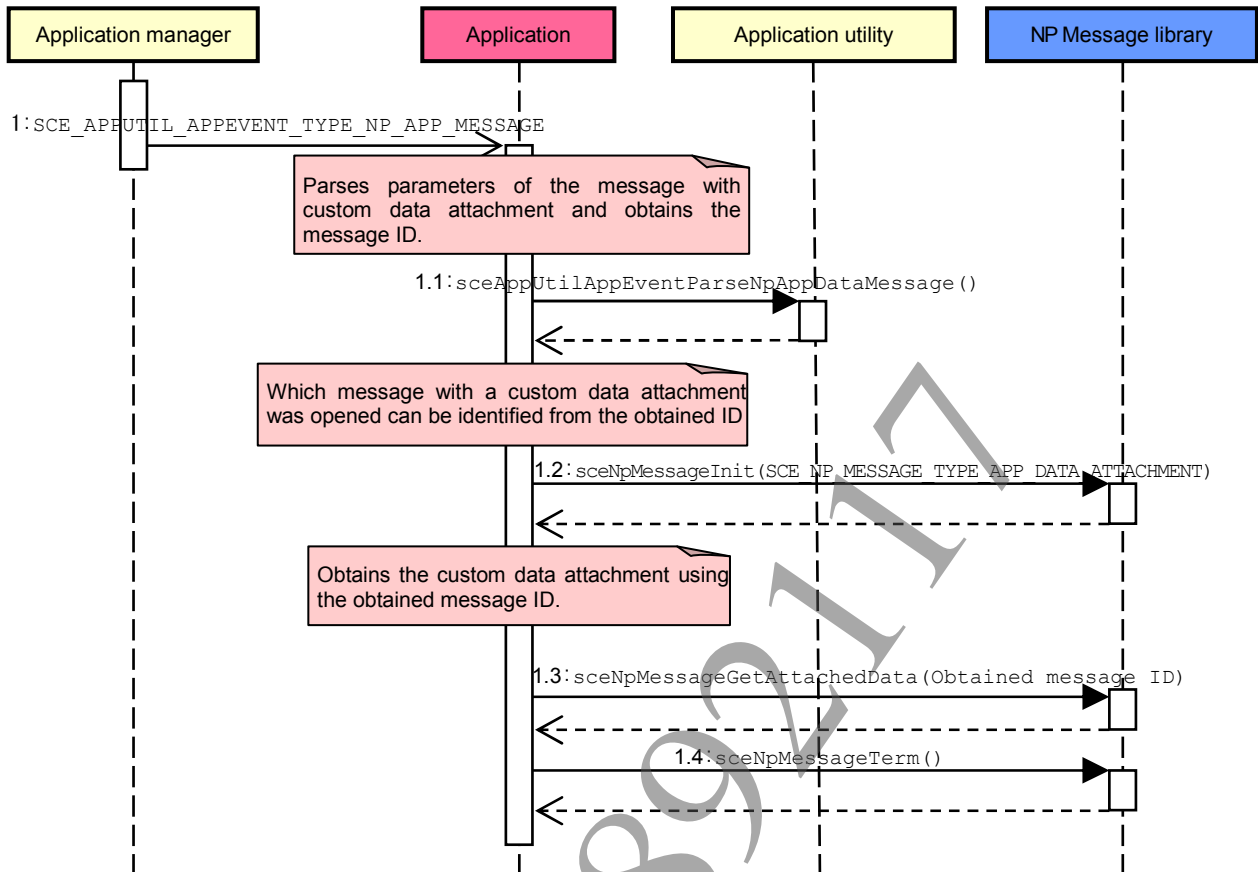


Figure 3 Procedure to Obtain Message with Custom Data Attachment Sent from the NP Message Feature

3 Reference Information

Starting an Application from the Messages Application

The user can open "game alert" from the Messages application to confirm received game custom data and game boot messages (messages with custom data attachments).

When the user opens a message from "game alert", the applicable application will start up.

The `SCE_APPUTIL_APPEVENT_TYPE_GAME_CUSTOM_DATA` event or `SCE_APPUTIL_APPEVENT_TYPE_NP_APP_DATA_MESSAGE` event will be sent from the application manager to the application immediately after application startup. Game custom data can subsequently be obtained in the same manner as described in the "Procedure for Obtaining Game Custom Data" section.

Display of Game Icons and Title Names for Titles Under Development

In the lists, detail screens, etc. that are displayed by the system software and game custom data 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 game custom data dialog, etc. can be obtained from the server of PSNSM, but application information for titles under development is not registered on the server of PSNSM.

Application information is uploaded to the server of PSNSM 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 PSNSM.