Message Dialog Overview

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Document serial number: 000004892117

Document serial number: 000004892117

Table of Contents

1 Library Overview

Purpose and Features

The Message Dialog library is a library that supports the display of information for users by applications. Through the use of the Message Dialog library, applications can easily implement guidance for users that complies with TRC (Technical Requirements Checklist).

The Message 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 display contents, monitoring the closing of the dialog through polling, and last obtaining the call results.

Main Features

The followings are the main features provided by the Message Dialog library.

- Feature to display user specified messages
- Feature to display system defined messages
- Feature to display error codes
- Feature to display progress bar

Embedding in Program

Include message_dialog.h in the source program. (Additionally, a number of header files are automatically included.)

The PRX module need not be loaded.

When building programs, link libSceCommonDialog_stub.a.

Sample Programs

The following files are provided as sample programs that use the Message Dialog library for reference purposes.

sample_code/system/api_message_dialog/user_message/

This sample displays user specified messages.

sample_code/system/api_message_dialog/system_message/

This sample displays system defined messages.

System defined messages include notifications of the availability of update files and information on magnetometer sensor calibration.

For the sample program displaying notification of the availability of update files, refer to the "GameUpdate Library Overview" document.

For the sample program displaying the information on magnetometer sensor calibration, refer to the "libmotion Overview" or "liblocation Overview" document.

sample_code/system/api_message_dialog/error_code/

This sample displays error codes.

sample_code/system/api_message_dialog/progress_bar/

This sample displays progress bar.

Reference Materials

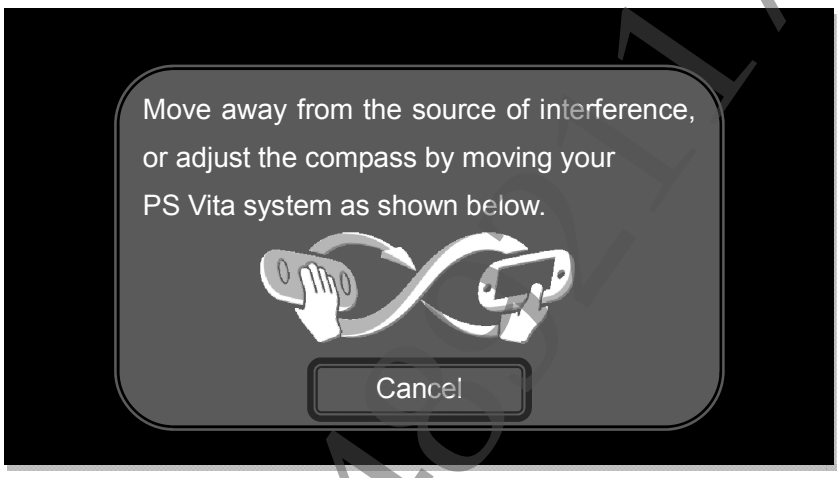
For the common limitations, specifications, etc. of the Common Dialog library, refer to the following document.

- Common Dialog Overview

Message on the Calibration of the Magnetometer Sensor

One of the features for displaying the system defined message is the feature for prompting the user to calibrate the magnetometer sensor.

Figure 1 Display of Information on Magnetometer Sensor Calibration



This display feature can be used only when using the libraries relating to magnetometer sensor, namely libmotion and liblocation, according to the usage method of each library.

For details, refer to the following document.

- Programming Startup Guide

Messages on the obtainment of Location Information

The feature to display system defined messages includes the feature to display messages regarding obtained location information.

There is a message displayed while location information is being obtained, and another message to announce the failure of obtaining location information.

Especially regarding the latter, several causes of failing to obtain location information using the liblocation library and their solutions exist according to the PlayStation®Vita model (3G/Wi-Fi) and Wi-Fi settings (**On**/**Off**). The feature to display a message upon failing to obtain location information automatically selects the appropriate message to display according to the situation. Thus, the application can use this display feature to indicate an appropriate procedure for the user to take upon failing to obtain location information without having to be aware of complex conditions.

There is a message with an **OK** button and a message with a **Cancel**/**Retry** button. Call the appropriate one according to the application's behavior upon failing to obtain location information.

Figure 2 Message while Obtaining Location Information

Figure 3 Message upon Failing to Obtain Location Information

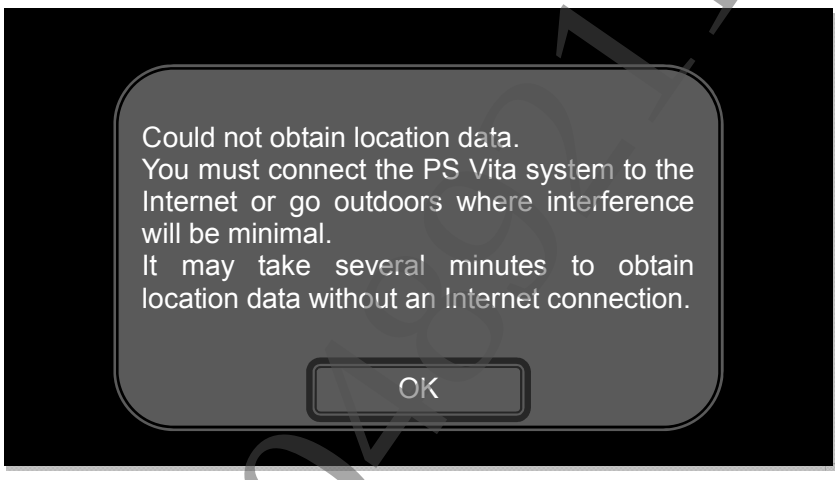
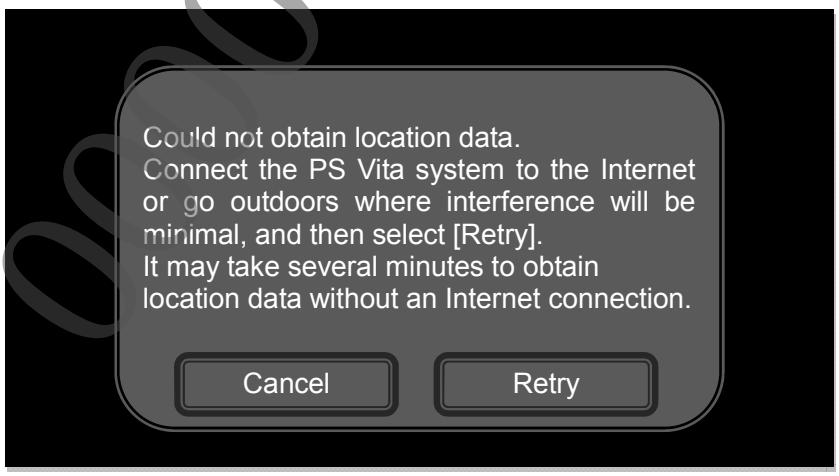


Figure 4 Message upon Failing to Obtaining Location Information (with a Retry)



Messages on the Availability of Update Files

The features for displaying system defined messages include a feature for displaying messages notifying the availability of application update files.

If an update file for the relevant application is available, it can be in one of the following three statuses:

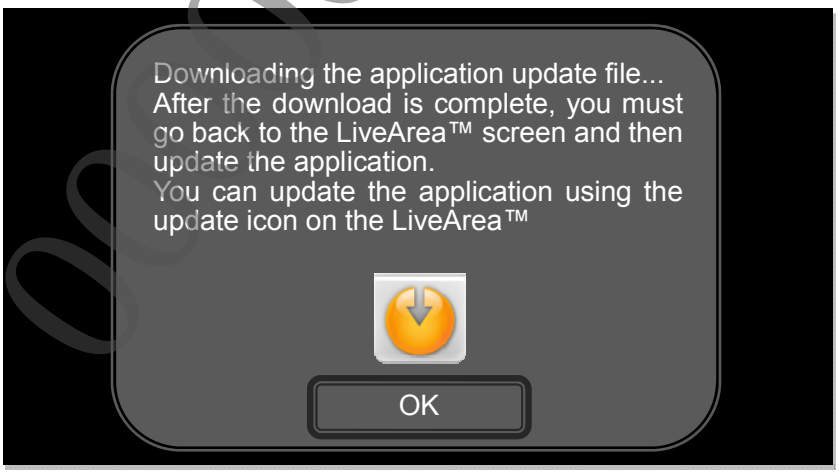
- Not yet downloaded
- Download in progress
- Downloaded

This feature detects status as necessary, and changes the contents of the messages displayed accordingly.

Note that this feature does not check whether application update files are available on the patch server; therefore, make sure to check the availability of application update files separately, and to only call this feature when an update file is available. Refer to the "GameUpdate Library Overview" document on how to check the availability of application update files.

Figure 5 Notification of the Availability of an Update File (Not Yet Downloaded/Downloaded)

Figure 6 Notification of the Availability of an Update File (Download in Progress)



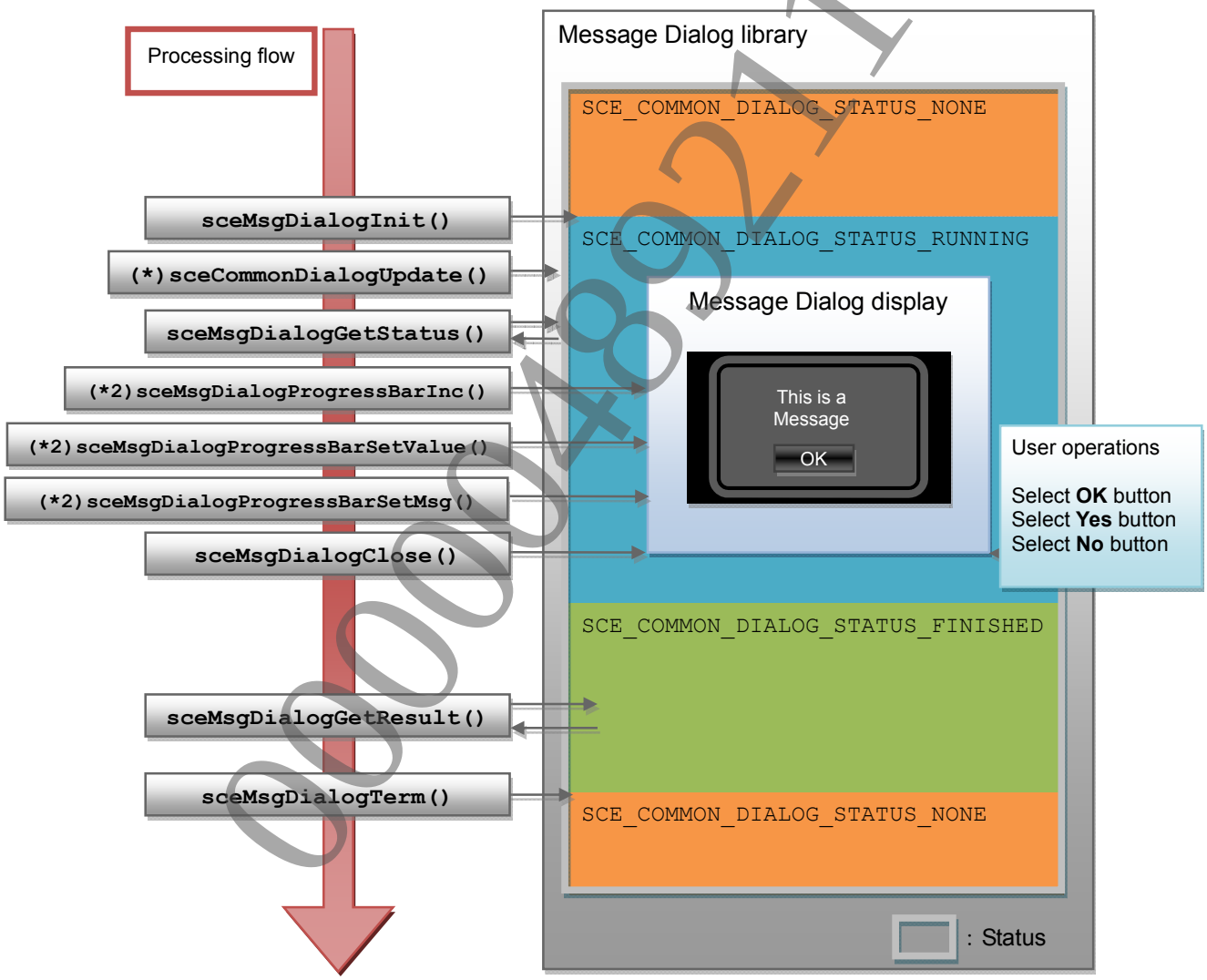
2 Usage Procedure

Basic Usage Procedure

The basic procedure to call the Message Dialog library is described below. The processing flow is outlined below.

(1) Set the parameters to the variables of the `SceMsgDialogParam` type.
(2) Initialize the library.
(3) Wait until Message Dialog is closed while updating the state as appropriate.
(4) Obtain the call results.
(5) End processing.

Figure 7 Basic Processing Procedure



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

`(*2)` These functions are available only when `SCE_MSG_DIALOG_MODE_PROGRESS_BAR` is being called.

Message Dialog Call Procedure

First, prepare the `SceMsgDialogParam` type variable and following initialization with `sceMsgDialogParamInit()`, set the operation mode (*mode*) and the parameters that are required accordingly.

(1) Initializing the library

Call `sceMsgDialogInit()` to perform initialization. Specify the `SceMsgDialogParam` type variable set beforehand as the argument.

(2) Obtaining the operation status

Call `sceMsgDialogGetStatus()` to poll the operation status of Message Dialog at every frame. `SCE_COMMON_DIALOG_STATUS_RUNNING` is returned as the operation status while Message Dialog is displayed, so wait until `SCE_COMMON_DIALOG_STATUS_FINISHED` is returned.

If Message Dialog without a button or a progress bar is displayed, call `sceMsgDialogClose()` at an arbitrary timing to close Message Dialog from the application.

Note
`sceCommonDialogUpdate()` must be called at every frame while the operation status is `SCE_COMMON_DIALOG_STATUS_RUNNING`. For details, refer to the "Common Dialog Overview" document.

(3) Updating the display information

While the operation status is `SCE_COMMON_DIALOG_STATUS_RUNNING`, the display information of Message Dialog can be updated as appropriate by calling the dedicated APIs additionally.

(4) Obtaining the call results

When Message Dialog closes and the operation status changes to `SCE_COMMON_DIALOG_STATUS_FINISHED` as a result, the results can be obtained with `sceMsgDialogGetResult()`. The results that can be obtained include the operation mode (*mode*) at initialization, return code, and selected button.

(5) Terminating the processing

Once the call results have been obtained, call `sceMsgDialogTerm()` to terminate the processing. As a result, the resources acquired during initialization are released.

Aborting the processing

When quitting an application, etc. to abort the display of Message Dialog from the application on an emergency basis, call `sceMsgDialogAbort()`. This ends the display faster than `sceMsgDialogClose()`, and `SCE_COMMON_DIALOG_RESULT_ABORTED` is returned as the return code.

Main APIs Used for Basic Processing

API	Description
`SceMsgDialogParam`	Parameter structure such as mode setting
`sceMsgDialogParamInit()`	Initializes parameter structure
`sceMsgDialogInit()`	Initializes library
`sceMsgDialogGetStatus()`	Obtains operation status
`sceMsgDialogGetResult()`	Obtains call results
`sceMsgDialogTerm()`	Closes library

3 Display Mode

User Specified Message Display Mode

- Applications can freely specify character strings to be displayed.
- The buttons to be displayed can be chosen from the following:
  - **OK** button
  - **Yes** and **No** buttons
  - No button
  - **OK** and **Cancel** buttons
  - **Cancel** button
  - User-specified character string button (types with 3 buttons only)
- The purpose for which this is to be used is freely selectable by the application.
- When using a user-specified character string button, it is possible to specify the size of the fonts on the button. Decrease character size in cases where the character string is too long and sticks out of the button.

System Defined Message Display Mode

- The messages previously prepared for the system are displayed.
- General purpose wordings and wordings for TRC compliance are provided as messages.
  Example "Please wait..."
  "Insufficient memory card capacity"
- The display language setting of the main unit is automatically reflected.
- Buttons cannot be specified. They are determined uniquely by the message type.

Error Code Display Mode

- The short error code for end users is displayed. For details on the short error code, refer to the "Error Overview" document.
- Call Message Dialog with the specification of a hexadecimal error code returned by the SDK library.
- Some of the error codes are automatically replaced with wordings that explain the error details.
- Display character strings cannot be specified.
- Buttons cannot be specified.

Progress Bar Display Mode

- Progress rate is displayed using the '%' label and an animated progress bar.
- Either the messages previously prepared for the system or the character strings freely specified by an application can be selected and displayed.
- Buttons cannot be specified. (Buttons are not displayed at all times.)
- Progress rate can be updated using `sceMsgDialogProgressBarInc()` and `sceMsgDialogProgressBarSetValue()` after displaying progress bar.
- The character strings can be updated using `sceMsgDialogProgressBarSetMsg()` after displaying progress bar.

User Operations and Call Results

Users can perform the following operations in relation to Message Dialog.

Figure 8 Message Dialog with an OK Button

Figure 9 Message Dialog with Yes and No Buttons

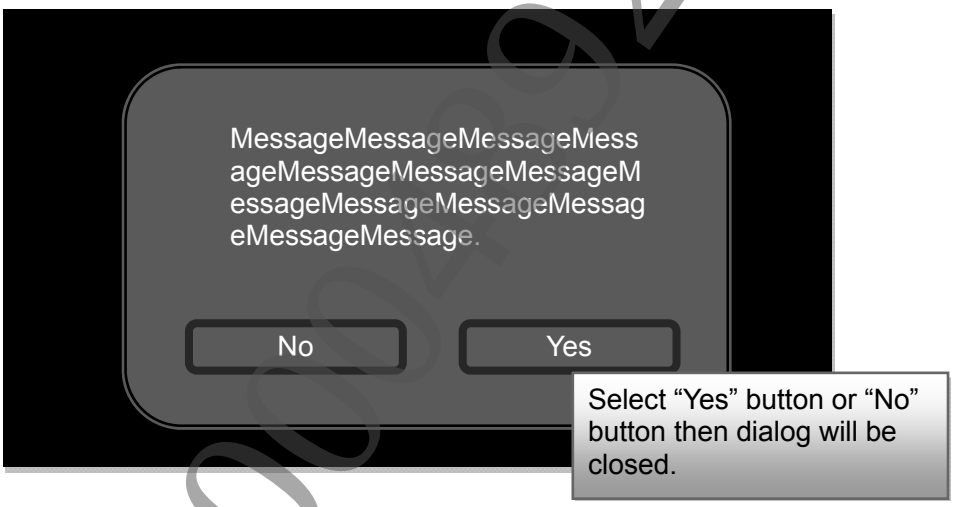


Figure 10 Message Dialog without a Button

Figure 11 Message Dialog without a Button (Small Size)

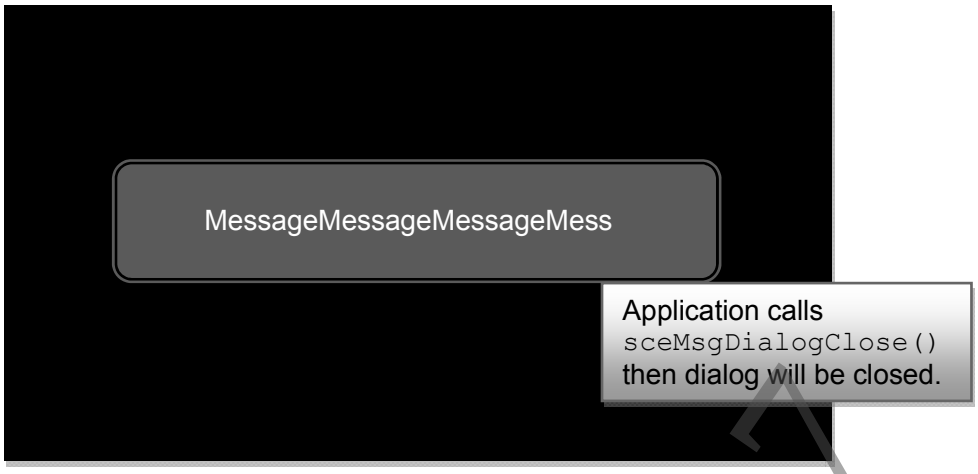


Figure 12 Progress Bar Dialog

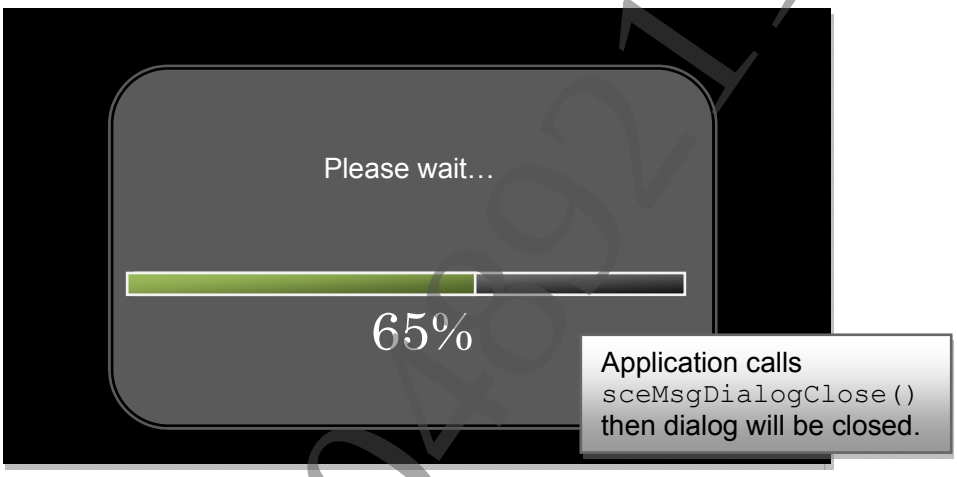


Figure 13 Message Dialog with a Cancel Button

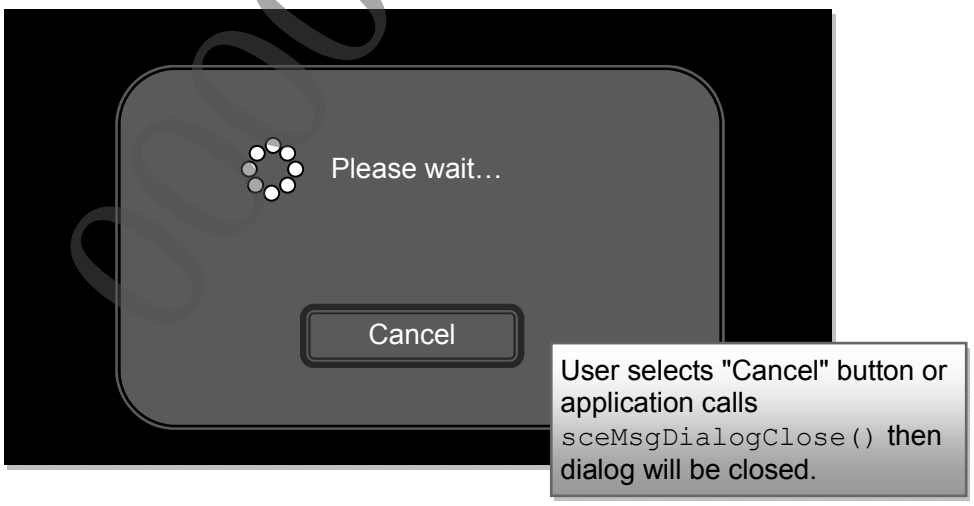
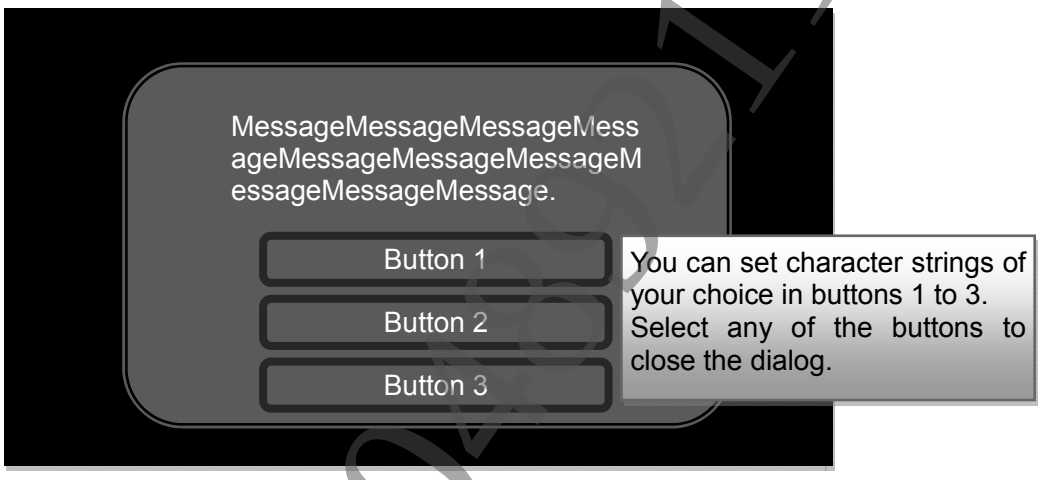


Figure 14 Message Dialog with OK and Cancel Buttons

Figure 15 Message Dialog with 3 Buttons



The application can obtain the Message Dialog call result to the `SceMsgDialogResult` structure using `sceMsgDialogGetResult()`.

Example:

- The user selected a button to close Message Dialog
  Call Result: Success (0)
  Button ID: **OK** button (`SCE_MSG_DIALOG_BUTTON_ID_OK`)
  Yes button (`SCE_MSG_DIALOG_BUTTON_ID_YES`)
  No button (`SCE_MSG_DIALOG_BUTTON_ID_NO`)
  Button 1 button (`SCE_MSG_DIALOG_BUTTON_ID_BUTTON1`)
  Button 2 button (`SCE_MSG_DIALOG_BUTTON_ID_BUTTON2`)
  Button 3 button (`SCE_MSG_DIALOG_BUTTON_ID_BUTTON3`)
- The user selected the **Cancel** button to close Message Dialog
  Call Result: Cancelled (`SCE_COMMON_DIALOG_RESULT_USER_CANCELED`)
  Button ID: None (`SCE_MSG_DIALOG_BUTTON_ID_INVALID`)
- The application closed Message Dialog (closed with `sceMsgDialogClose()`)
  Call Result: Success (0)
  Button ID: None (`SCE_MSG_DIALOG_BUTTON_ID_INVALID`)

- The application aborted Message Dialog calling (closed with `sceMsgDialogAbort()`)
  Call Result: Aborted (`SCE_COMMON_DIALOG_RESULT_ABORTED`)
  Button ID: None (`SCE_MSG_DIALOG_BUTTON_ID_INVALID`)

4 System Defined Messages for TRC

Some TRC requirements define that it is necessary to display an appropriate message according to the operation status of the application. Through system defined message display mode, such appropriate message can be displayed.

Guidance Display when Microphone Is Disabled

This display feature is used in order to satisfy the TRC requirement R3086.

The content of this guidance display will automatically be switched depending on whether a microphone is detected as a device or not. The application cannot specify which to display.

Either one of the messages, Figure 16 or Figure 17, can be used.

Figure 16 Guidance Display A when Microphone Is Disabled (with a Microphone Detected)

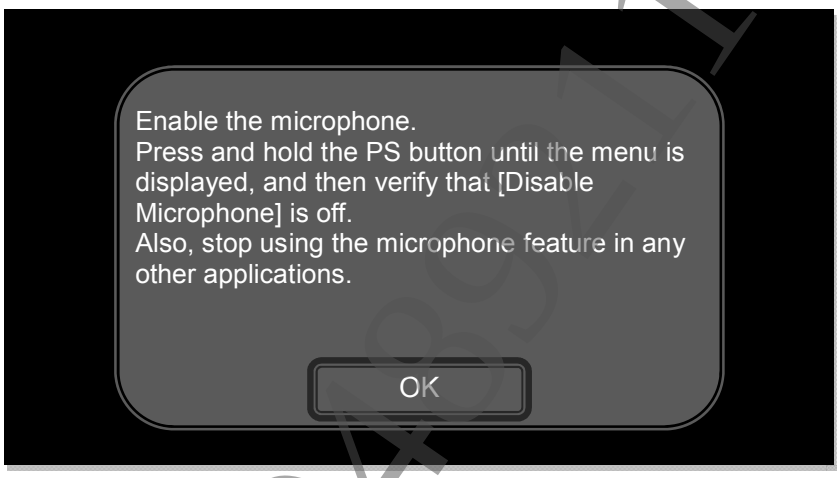


Figure 17 Guidance Display B when Microphone Is Disabled (with a Microphone Detected)

Figure 18 Guidance Display when Microphone Is Disabled (with a Microphone Undetected)

Guidance Display when Operation Requires Wi-Fi Connection

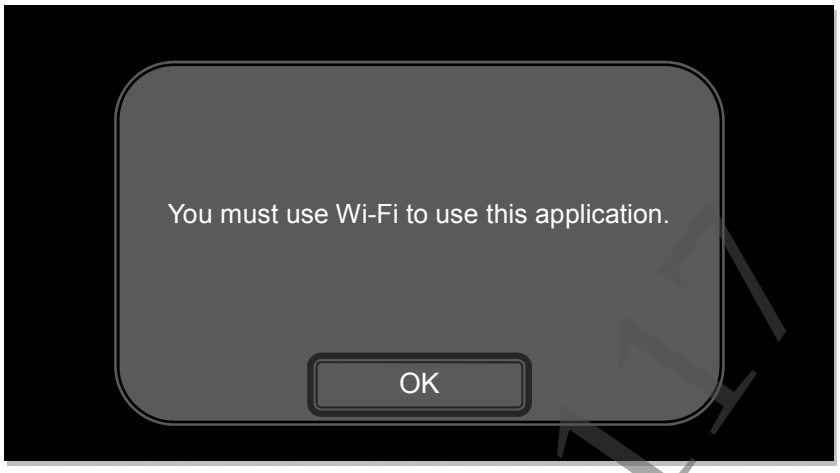
This display feature is used in order to satisfy the TRC requirements R3132 and 3134.

Figure 19 Guidance Display when Operation Requires Wi-Fi Connection

Guidance Display when Application Requires Wi-Fi Connection

This display feature is used in order to satisfy the TRC requirement R3132.

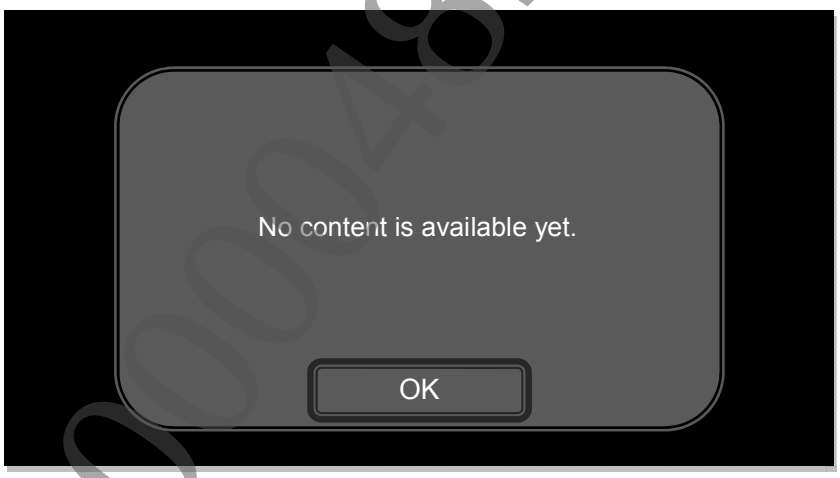
Figure 20 Guidance Display when Application Requires Wi-Fi Connection



Guidance Display when No Content Is Available in the Store

This display feature is used in order to satisfy the TRC requirement R3125.

Figure 21 Guidance Display when No Content Is Available in the Store



Guidance Display when PSN℠ Features Cannot Be Used Due to Age Restrictions

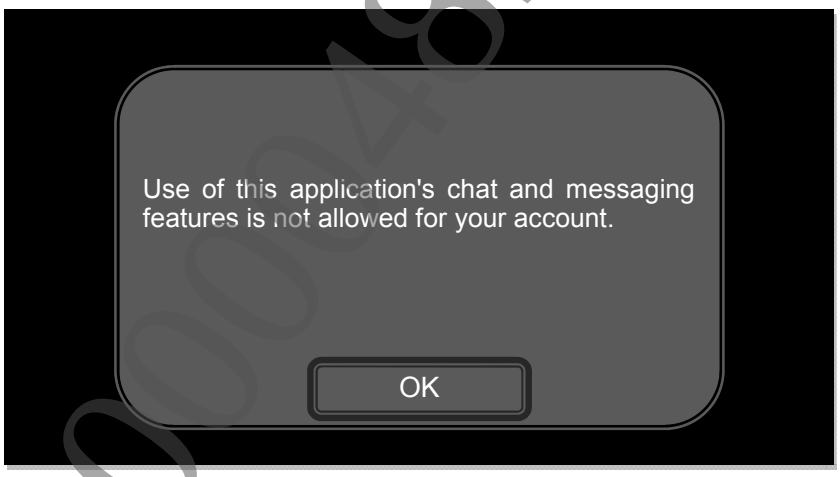
This display feature is used in order to satisfy the TRC requirement R3052.

Figure 22 Guidance Display when PSN℠ Features Cannot Be Used Due to Age Restrictions

Guidance Display when Use of Chat and Messaging Features Is Not Allowed

This display feature is used in order to satisfy the TRC requirement R3053.

Figure 23 Guidance Display when Use of Chat and Messaging Features Is Not Allowed



5 Precautions

Limitations

Common Dialog limitations apply.