Camera Import Dialog Overview

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Table of Contents

1 Library Overview

Purpose and Features

The Camera Import Dialog library provides a function for taking photographs with a UI. By using the Camera Import Dialog library, the application can easily implement processing for displaying the photography screen and taking photographs.

The Camera Import Dialog library is one of the functions in the Common Dialog library. It conceals GUI display and user operations handling.

With the Camera Import Dialog library, the data of the photographs taken is not recorded to the system software. If you wish to register the data of the photographs taken to the system software, do so by using the Photo Export library.

Main Functions

The main functions offered by Camera Import Dialog are as follows:

- Displaying the photography screen
- Providing photographed data to the application
- Setting/switching camera device (front/rear)
- Setting/switching camera resolution
- Display of overlay images when taking photographs and superimposition function when saving

Embedding into a Program

Include cameraimport_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 Programs

The following program is provided as a Camera Import Dialog sample program for reference purposes.

sample_code/system/api_cameraimport/fixed_basic/

This sample program shows the basic usage of Camera Import Dialog.

It demonstrates how to implement processing for taking photographs with Camera Import Dialog and registering the data of these photographs to the system software using the Photo Export library. It also features a usage example of the overlay function.

Reference Materials

For the 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 Camera Import Dialog library is described below. The processing flow is outlined below.

(1) Set the parameters to the variables of the `SceCameraImportDialogParam` type.

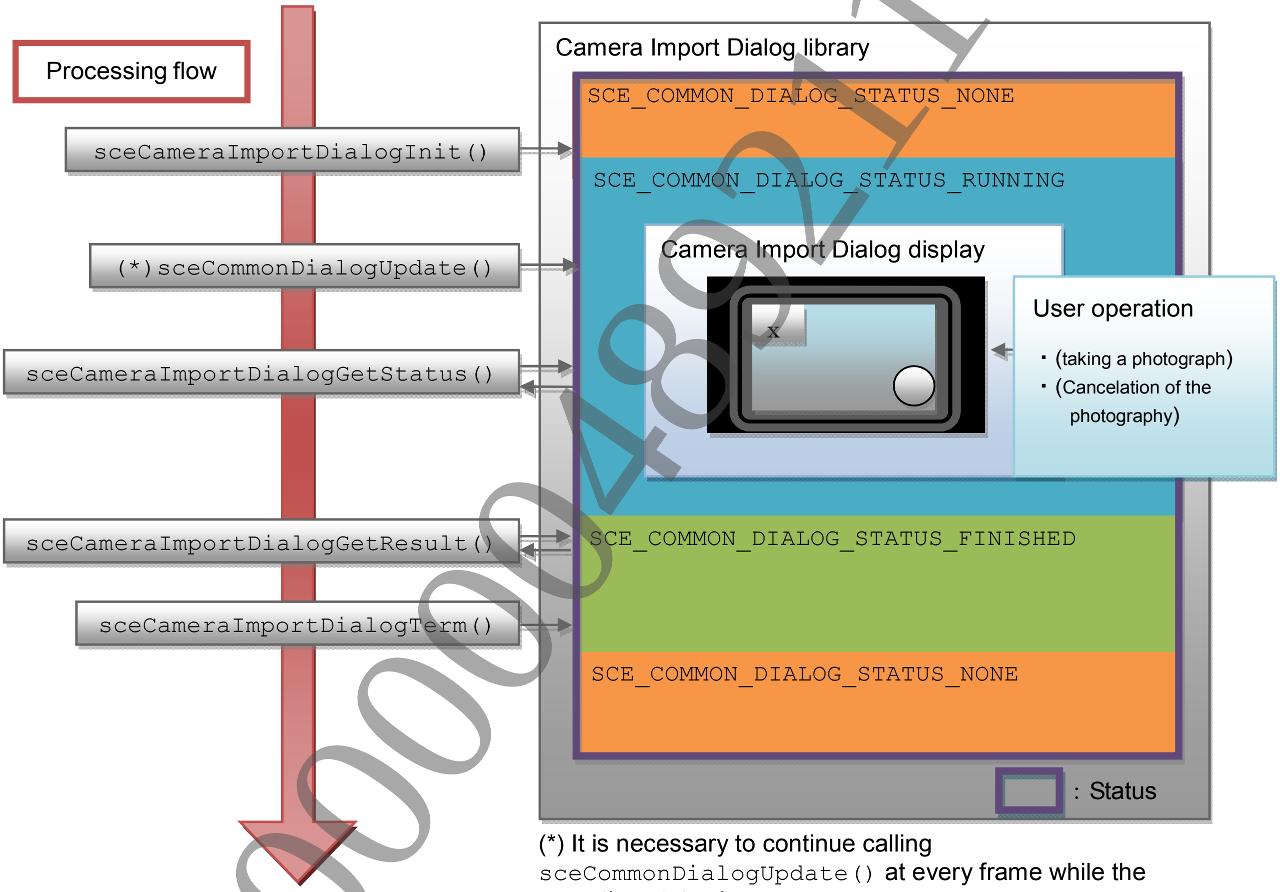
(2) Call a function.

(3) Wait for the response from the dialog.

(4) Retrieve the call results.

(5) End processing.

Figure 1 Basic Processing Procedure


Processing flow
sceCameraImportDialogInit()
(*)sceCommonDialogUpdate()
sceCameraImportDialogGetStatus()
sceCameraImportDialogGetResult()
sceCameraImportDialogTerm()
Camera Import Dialog library
SCE_COMMON_DIALOG_STATUS_NONE
SCE_COMMON_DIALOG_STATUS_RUNNING
Camera Import Dialog display
x
User operation
・(taking a photograph)
・(Cancelation of the photography)
SCE_COMMON_DIALOG_STATUS_FINISHED
SCE_COMMON_DIALOG_STATUS_NONE
: Status

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

Camera Import Dialog Call Procedure

(1) Prepare the operation parameters

First, prepare the `SceCameraImportDialogParam` type variable and following initialization with `sceCameraImportDialogParamInit()`, be sure to set the parameters that are required accordingly.

Parameters include the camera resolution, camera devices (front/rear) and photograph data format available to the users. Also, a working memory is necessary for Camera Import Dialog. This working memory also needs to be specified here.

(2) Calling the function

Call a Camera Import Dialog function with `sceCameraImportDialogInit()`. Specify the `SceCameraImportDialogParam` type variable set beforehand as the argument.

(3) Waiting for the response from the dialog

Call `sceCameraImportDialogGetStatus()` to poll the operation status of Camera Import Dialog at each frame.

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.

(4) Retrieving the call result

When the operation status changes to `SCE_COMMON_DIALOG_STATUS_FINISHED`, the results can be retrieved with `sceCameraImportDialogGetResult()`. The results that can be retrieved include the result of user operation (photograph taken/cancelled) and supplementary information on photographed image data. Image data is stored in the location specified with `SceCameraImportDialogParam` when the function is called.

(5) Terminating the processing

When the operation status becomes `SCE_COMMON_DIALOG_STATUS_FINISHED`, call `sceCameraImportDialogTerm()` to terminate the processing. As a result, the resources acquired during calling are released, and the operation status becomes `SCE_COMMON_DIALOG_STATUS_NONE`.

Aborting the Processing

When quitting an application, etc., to abort the display of Camera Import Dialog from the application side on an emergency basis, call `sceCameraImportDialogAbort()`.
`SCE_COMMON_DIALOG_RESULT_ABORTED` is returned as a return code.

Main APIs Used for Basic Processing

API	Description
`SceCameraImportDialogParam`	Parameter structure such as operation mode setting
`sceCameraImportDialogParamInit()`	Initializes parameter structure
`sceCameraImportDialogInit()`	Calls function
`sceCameraImportDialogGetStatus()`	Retrieves operation status
`sceCameraImportDialogGetResult()`	Retrieves call results
`sceCameraImportDialogTerm()`	Ends calling of function
`sceCameraImportDialogAbort()`	Aborts calling of function

3 Reference Information

Libraries to Be Loaded and Initialized Beforehand

In order to use the Camera Import Dialog library, the dependent library must be loaded by calling the following API beforehand.

```
sceSysmoduleLoadModule(SCE_SYSMODULE_SHUTTER_SOUND);
```

Photograph Data Format

It is possible to select the output format for the data of the photographs taken.

RAW Format

When `SCE_CAMERAIMPORT_DIALOG_OUTPUT_MODE_MEMORY_RAW` is set in the *`outputMode`* of `SceCameraImportDialogParam`, the photographed data will be output in non-compressed format. Set the area for writing data with the *`buffer`* and *`bufferSize`* of `SceCameraImportDialogMemoryRawParam`, and specify the data format with *`texType`* and *`texFormat`*.

JPEG Format

When `SCE_CAMERAIMPORT_DIALOG_OUTPUT_MODE_MEMORY_JPEG` is set in the *`outputMode`* of `SceCameraImportDialogParam`, photographed data will be output in JPEG format. Specify the area for writing data with the *`buffer`* and *`bufferSize`* of `SceCameraImportDialogMemoryJpegParam`.

Vertical/Horizontal Photograph Direction Information

Camera Import Dialog provides a function for determining whether a photograph has been taken in a vertical or horizontal position by using acceleration sensor information. The information is stored in the *`orientation`* variable of `SceCameraImportDialogOutputParam`. Since, in the case of RAW format, the data stored in memory is not rotated, rotate it when using it with the application by making reference to *`orientation`* information. Images are not stored as rotated in JPEG format, either; however, since rotation information is stored in the Orientation Tag in JPEG data, the image will be adjusted in accordance with the vertical/horizontal position at the time the photograph was taken when displayed with the appropriate viewer during previews.

Also, if `SCE_CAMERAIMPORT_DIALOG_ROTATION_MODE_DISABLE` is specified in the *`rotationMode`* of `SceCameraImportDialogParam`, use of the acceleration sensor will be suspended inside Camera Import Dialog, and it will be possible to cancel the storage of rotation information to the *`orientation`* variable of `SceCameraImportDialogOutputParam` and to JPEG.

Overlay Images

Camera Import Dialog allows you to overlay arbitrary images on the photography screen and on photograph data. Perform overlay instructions with the *`overlayImage`* and *`overlayMode`* of `SceCameraImportDialogParam`.

Limitations

There is no function available to adjust overlay position accordingly if the user switches between vertical and horizontal holding position. If using the overlay function, we recommend that you do not allow the use of rotation information (set with *`rotationMode`*).

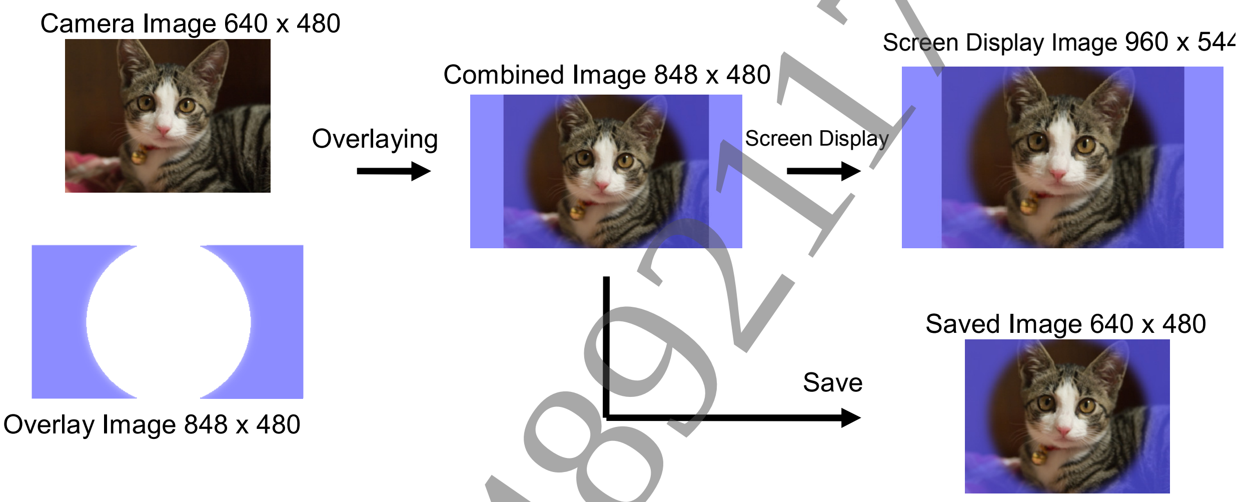
Overlaying Position

Specified overlay images are positioned according to the following logic:

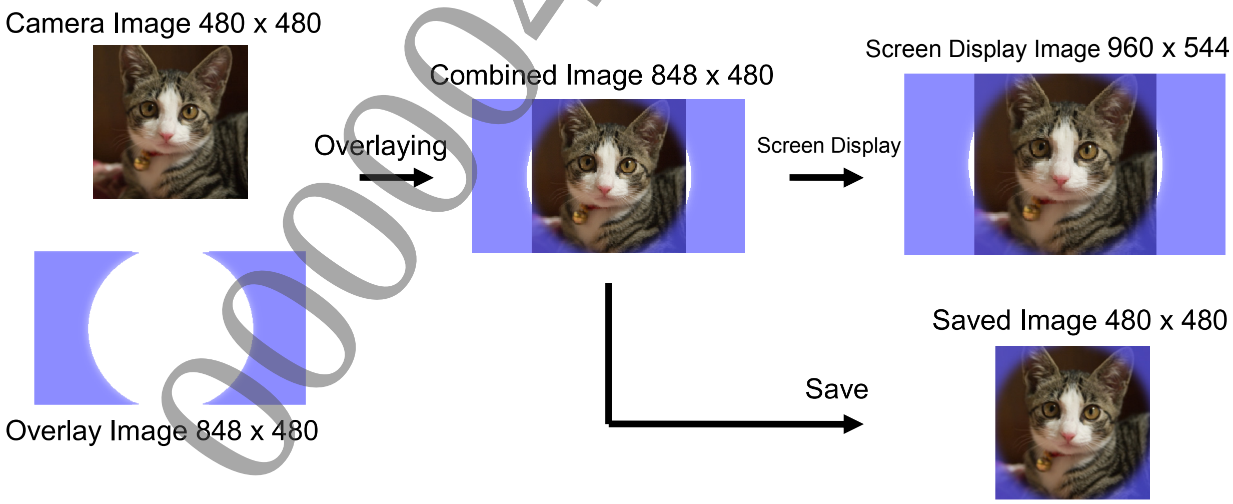
- Combined images composed by camera images and overlay images
  First, combined images overlaying camera images and overlay images dot by dot are generated internally. They will be positioned centrally, both horizontally and vertically.
- When used for on-screen display
  Combined images are stretched when displayed so that the camera image part fits the screen.
- When used as photograph data
  The camera image part of combined images will be saved as a clip.

Below are examples of overlaying for each level of camera resolution (640 x 480, 480 x 480, 640 x 360).

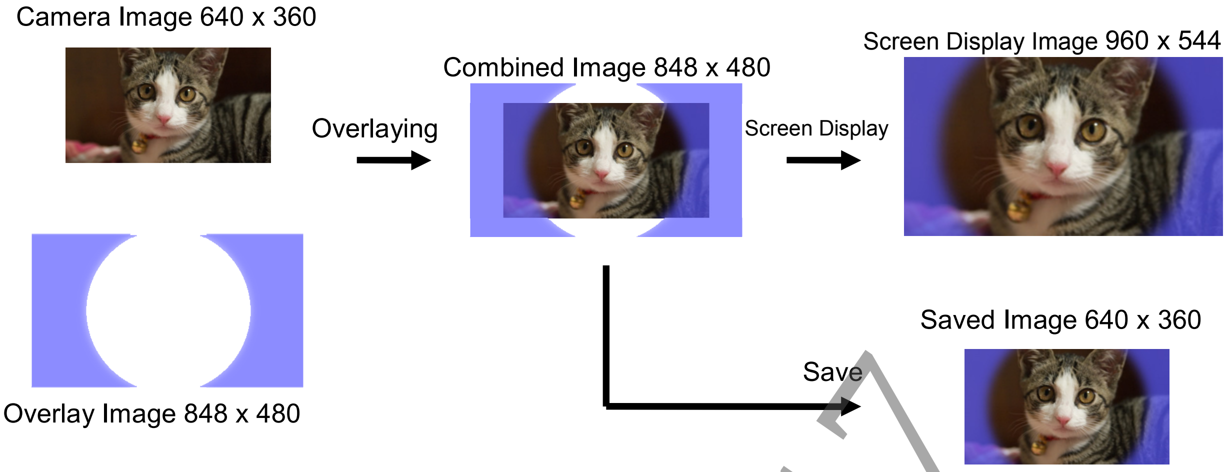
(Example) When camera resolution is 640 x 480:



(Example) When camera resolution is 480 x 480:



(Example) When camera resolution is 640 x 360:



4 Limitations

Mutually Exclusive Use of libcamera and Camera Import Dialog Library

In a process using Camera Import Dialog, if the libcamera is in use the processing of Camera Import Dialog will fail. If the libcamera is used, execute `sceCameraClose()` before calling `sceCameraImportDialogInit()` to end the use of the libcamera. Also, do not call any of the APIs of the libcamera while using Camera Import Dialog.

Power Configuration Setting

Camera Import Dialog uses the camera devices internally. If `scePowerSetConfigurationMode()` is used in a process, set a mode in which the camera devices can be used before calling `sceCameraImportDialogInit()`, and do not change the mode while Camera Import Dialog is in use.

If the camera device cannot be used, the `SCE_CAMERAIMPORT_DIALOG_ERROR_REQUIRED_DEVICE_CANNOT_USE` error is returned to the *`result`* variable of the `SceCameraImportDialogResult` structure obtained with the `sceCameraImportDialogGetResult()` function.

For details on `scePowerSetConfigurationMode()` and power configuration control, refer to the "Power Service Overview" and "Power Service Reference".

Working Buffer

The working buffer given with the *`workingBuffer`* of `SceCameraImportDialogParam` must be a continuous physical memory, uncached and with 4 KiB alignment. Also, memory must be mapped so as to allow read access from the GPU. Memory mapping can be achieved by calling `sceGxmMapMemory()` with `SCE_GXM_MEMORY_ATTRIB_READ` specified. For details, refer to the "libgxm Reference" document.

The necessary size of the working buffer is defined by the constant `SCE_CAMERAIMPORT_DIALOG_WORKING_BUFFER_SIZE`.

Handling of Location Information

Camera Import Dialog does not handle location information. GPSInfo IFD will not be written even if photographic results are output in JPEG format.

Other Limitations

Common Dialog limitations apply.