Video Import Dialog Overview

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Table of Contents

Document serial number: 000004892117

1 Library Overview

Scope of This Document

This document explains the video import dialog library, which supports the import of video data by applications. The following features are realized by the video import dialog library.

- Feature to display video list
- Feature to select one video file

Purpose and Features

The video import dialog library is a library to support the realization of video data import by applications. Applications can easily implement the processing to display a video list and select video data through the use of the video import dialog library.

The video import dialog library is one of the features in the Common Dialog library. It conceals GUI display and user operations handling. The usage flow is as follows: first, call specifying the dialog call mode and selection target category; then, monitor shut-down of the dialog by polling; finally, obtain call results.

Main Features

The main features offered by the video import dialog are as follows:

- Feature to display video list
- Feature to select one video file

Embedding into a Program

Include videoimport_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 video import dialog sample program for reference purposes.

sample_code/system/api_videoimport_dialog/fixed_basic/

This sample uses the features of the video import dialog to execute video file selection.

sample_code/system/api_videoimport_dialog/playback_on_avplayer/

This sample uses the video player library to play back the content selected in the video import dialog.

Reference Materials

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

- Common Dialog Overview

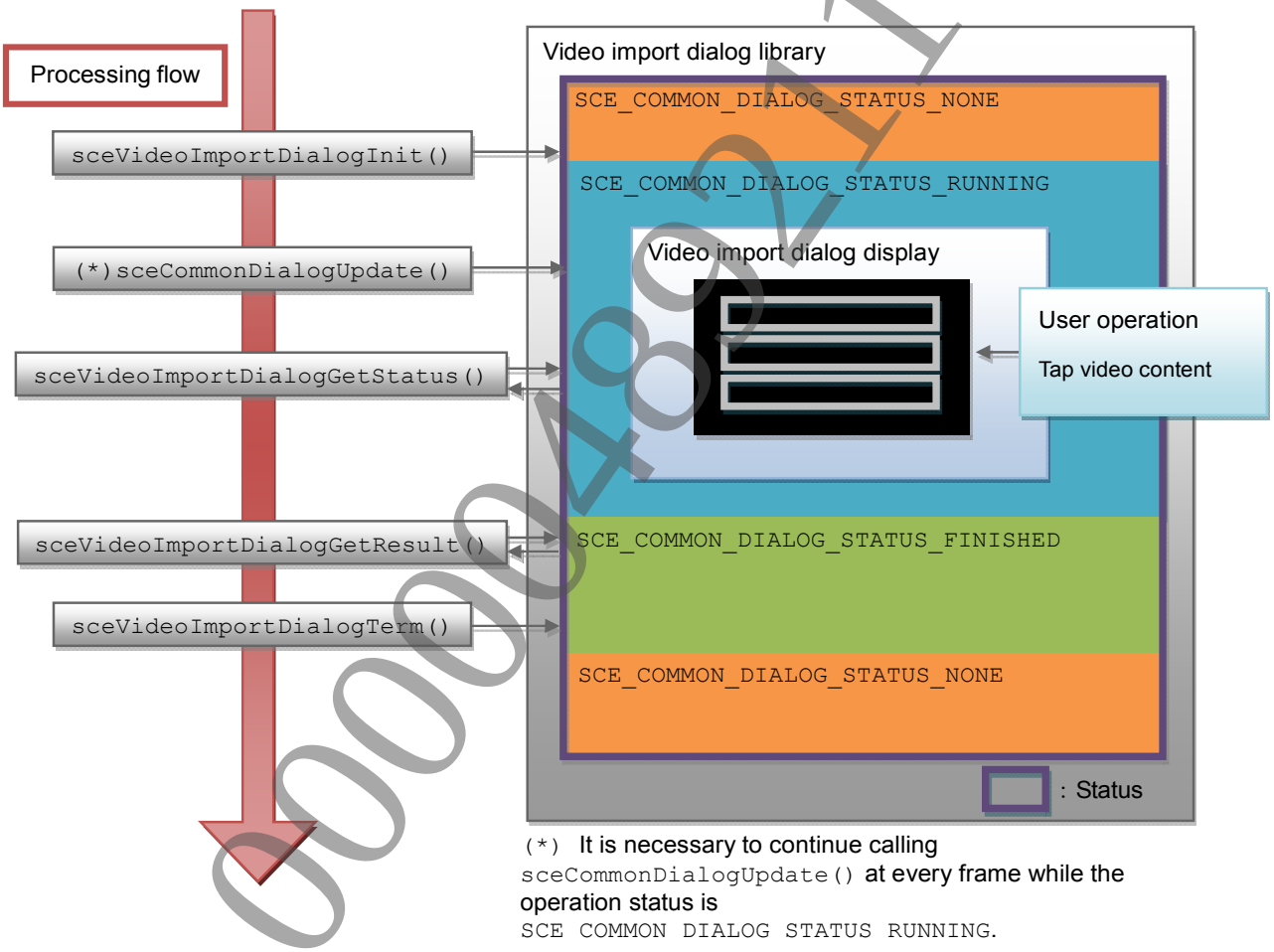
2 Usage Procedure

Basic Usage Procedure

The basic procedure to call the video import dialog library is described below. The processing flow is outlined below.

(1) Set the parameters to the variables of the `SceVideoImportDialogParam` type.
(2) Call a feature.
(3) Wait for the response from the dialog.
(4) Obtain the call results.
(5) End processing.

Figure 1 Basic Processing Procedure



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

Video Import Dialog Call Procedure

First, prepare the `SceVideoImportDialogParam` type variable and following initialization with `sceVideoImportDialogParamInit()`, be sure to set the dialog call mode (*`mode`*) and the parameters that are required accordingly. For details on each parameter, refer to the "`SceVideoImportDialogParam`" section in the "Video Import Dialog Reference" document.

(1) Calling the feature

Call a video import dialog feature with `sceVideoImportDialogInit()`. Specify the `SceVideoImportDialogParam` type variable set beforehand as the argument.

(2) Waiting for the response from the dialog

Call `sceVideoImportDialogGetStatus()` to poll the operation status of the video 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.

(3) Obtaining the call results

When the operation status changes to `SCE_COMMON_DIALOG_STATUS_FINISHED`, the results can be obtained with `sceVideoImportDialogGetResult()`. The results that can be obtained include the selection result at the time of the call (whether the user made a selection, whether it was canceled), and the number of selected files. The information of the selected file(s) is set in *`itemData`* of `SceVideoImportDialogParam`.

(4) Terminating the processing

When the operation status becomes `SCE_COMMON_DIALOG_STATUS_FINISHED`, call `sceVideoImportDialogTerm()` 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 the video import dialog from the application side on an emergency basis, call `sceVideoImportDialogAbort()`. Display will quickly terminate, and operation status will change to `SCE_COMMON_DIALOG_STATUS_FINISHED`. In this case, too, the call result will be obtained with `sceVideoImportDialogGetResult()`. `SCE_COMMON_DIALOG_RESULT_ABORTED` is returned as obtained result.

Main APIs Used for Basic Processing

API	Description
`SceVideoImportDialogParam`	Parameter structure such as mode setting
`sceVideoImportDialogParamInit()`	Initializes parameter structure
`sceVideoImportDialogInit()`	Calls feature
`sceVideoImportDialogGetStatus()`	Obtains operation status
`sceVideoImportDialogGetResult()`	Obtains call results
`sceVideoImportDialogTerm()`	Ends calling of feature
`sceVideoImportDialogAbort()`	Aborts calling of feature

3 Reference Information

Display of Thumbnails in Dialog and Accessing Obtained File Paths

When thumbnail display is enabled in the video import dialog, `sceAppUtilExtVideoMount()` must be called in advance to mount the video0: device before calling `sceVideoImportDialogInit()`.

In addition, the video0: device must also be mounted when accessing file paths of obtained video data.

When accessing the video0: device is no longer required, call `sceAppUtilExtVideoUmount()` to unmount the video0: device.

Note
An error will occur if unmounting is attempted during video0: device access. Do not perform unmounting during video0: device access.

Preparations for `sceAppUtilExtVideoMount()`/`sceAppUtilExtVideoUmount()` Execution

`sceAppUtilExtVideoMount()`/`sceAppUtilExtVideoUmount()`, which perform video0: device mounting/unmounting, are provided by the application utility library.

To embed the application utility library into programs, include apputil_ext.h and link libSceAppUtilExt_stub.a or libSceAppUtilExt_stub_weak.a.

In addition, execute `sceSysmoduleLoadModule(SCE_SYSMODULE_APPUTIL_EXT)` in the program in advance to load PRX modules.

For details on the application utility library, refer to the following documents.

- Application Utility Overview
- Application Utility Reference

4 Precautions

Limitations

Common Dialog limitations apply.