

Store Checkout Dialog Overview

© 2014 Sony Computer Entertainment Inc.
All Rights Reserved.
SCE Confidential

Table of Contents

1 Library Overview.....	3
Scope of This Document.....	3
Purpose and Features.....	3
Main Functions	3
Embedding into a Program	3
Sample Programs.....	4
Reference Materials	4
2 Usage Procedure	5
Basic Usage Procedure	5
Store Checkout Dialog Call Procedure	6
3 Reference Information.....	7
Calling Modes.....	7
Funding the Wallet	8
Display Language.....	8
4 Precautions	9
Limitations	9

1 Library Overview

Scope of This Document

This document describes the Store Checkout Dialog library that enables applications to purchase and download (including installation) the products displayed in PlayStation®Store (the title store) and, to install the downloaded additional contents and to upgrade applications to the full version..

Purpose and Features

The Store Checkout Dialog library provides functions for purchasing and downloading (including installing) products from the PlayStation®Store (the title store). A product can be purchased and downloaded by first using the NP IN-GAME Commerce 2 library to get the product catalog. Information can be obtained on the desired product from the product catalog from where it is passed to the Store Checkout Dialog library. Multiple products can be purchased and downloaded at the same time. Moreover, if there are any additional contents and upgrade licenses that have already been downloaded, installation of the additional contents and upgrade to the full version can be performed collectively.

The Store Checkout Dialog library is provided as a library with a user interface. When the Store Checkout Dialog library is called, it displays various screens such as the Purchase Confirmation screen and the user can perform the actual purchasing and downloading operations. This library also provides a function for funding the wallet during purchase processing in addition to a download-only function for previously purchased products.

Main Functions

The main functions offered by Store Checkout Dialog are as follows:

- Function to purchase the specified products
- Function to download the specified products
- Function to install the downloaded additional contents and upgrade applications to the full version

Embedding into a Program

Include `store_checkout_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`.

The Store Checkout Dialog library calls the NP IN-GAME Commerce 2 library internally. This module set must be loaded and initialized in advance to make the NP IN-GAME Commerce 2 library available for use.

For information about loading and initializing the NP IN-GAME Commerce 2 library, refer to the "NP IN-GAME Commerce 2 Overview" and "NP IN-GAME Commerce 2 Reference" documents.

Sample Programs

The following program is provided as a Store Checkout Dialog sample program for reference purposes.

sample_code/network/api_np/np_gui_commerce2/

This is a comprehensive sample program of the NP IN-GAME Commerce 2 library and uses the Store Checkout Dialog functions.

Reference Materials

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

- Common Dialog Overview

For the specifications of the additional contents and details of the upgrade to the full version, refer to the following document.

- Application Development Process Overview

For the information on the LiveArea™ specifications in the case of the upgrade to the full version, refer to the following document.

- LiveArea™ Specifications

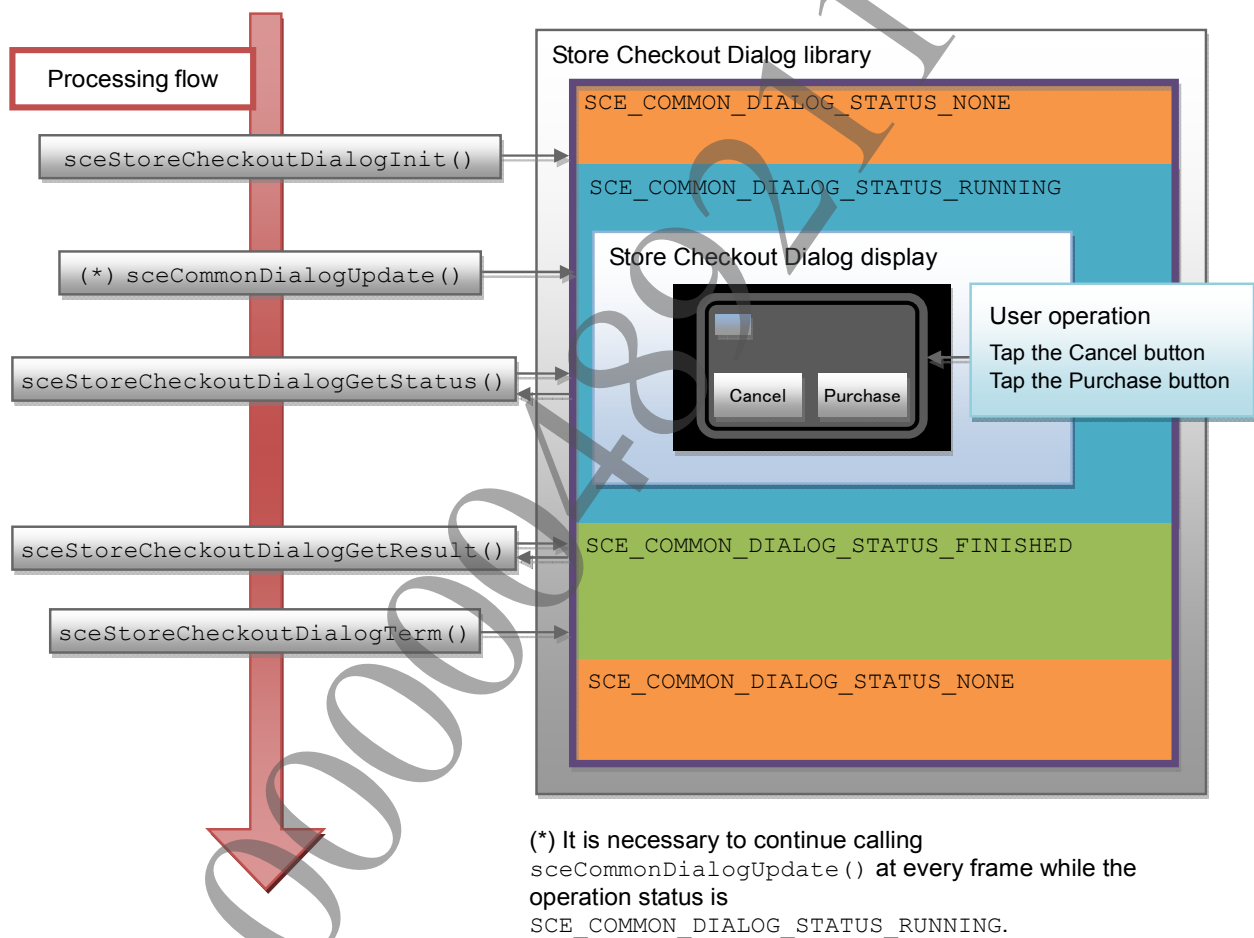
2 Usage Procedure

Basic Usage Procedure

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

- (1) Set the parameters to the variables of the `SceStoreCheckoutDialogParam` 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



Store Checkout Dialog Call Procedure

First, prepare the `SceStoreCheckoutDialogParam` type variable and following initialization with `sceStoreCheckoutDialogParamInit()`, be sure to set the operation mode (*mode*) and the parameters that are required accordingly.

(1) Calling the function

Call a Store Checkout Dialog function with `sceStoreCheckoutDialogInit()`. Specify the `SceStoreCheckoutDialogParam` type variable set beforehand as the argument.

If a directory or file under "addcont0:" is accessed, the `SCE_STORE_CHECKOUT_DIALOG_ERROR_ADDCONT_BUSY` error results. Call `sceStoreCheckoutDialogInit()` after terminating all accesses to "addcont0:" beforehand. If the additional contents directory is opened using `sceAppUtilDrmOpen()` but no file is accessed, calling `sceStoreCheckoutDialogInit()` is successful, but note that the additional contents opened using `sceAppUtilDrmOpen()` are automatically closed.

(2) Waiting for the response from the dialog

Call `sceStoreCheckoutDialogGetStatus()` to poll the operation status of Store Checkout 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) Retrieving the call results

When the operation status changes to `SCE_COMMON_DIALOG_STATUS_FINISHED`, the results can be retrieved with `sceStoreCheckoutDialogGetResult()`.

(4) Terminating the processing

When the operation status becomes `SCE_COMMON_DIALOG_STATUS_FINISHED`, call `sceStoreCheckoutDialogTerm()` 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

To abort the display of Store Checkout Dialog from the application side on an emergency basis, for example, when quitting the application, call `sceStoreCheckoutDialogAbort()`. When calling is successful, Store Checkout Dialog will begin to abort processing. When processing is completely aborted, the operation status will change to `SCE_COMMON_DIALOG_STATUS_FINISHED`. In this case, too, the call result will be retrieved with `sceStoreCheckoutDialogGetResult()`. `SCE_COMMON_DIALOG_RESULT_ABORTED` is returned as retrieved result.

Main APIs Used for Basic Processing

API	Description
<code>SceStoreCheckoutDialogParam</code>	Parameter structure including mode setting
<code>sceStoreCheckoutDialogParamInit()</code>	Initializes parameter structure
<code>sceStoreCheckoutDialogInit()</code>	Calls function
<code>sceStoreCheckoutDialogGetStatus()</code>	Retrieves operation status
<code>sceStoreCheckoutDialogGetResult()</code>	Retrieves call results
<code>sceStoreCheckoutDialogTerm()</code>	Ends calling of function
<code>sceStoreCheckoutDialogAbort()</code>	Aborts calling of function

3 Reference Information

Calling Modes

The Store Checkout Dialog library has the following calling modes.

Calling Mode	Description
SCE_STORE_CHECKOUT_DIALOG_MODE_CHECKOUT2	Startup in checkout mode
SCE_STORE_CHECKOUT_DIALOG_MODE_DOWNLOAD2	Startup in download mode
SCE_STORE_CHECKOUT_DIALOG_MODE_INSTALL_AND_UPGRADE	Startup in install and upgrade mode

Startup in Checkout Mode

In checkout mode, the dialog starts up in a state in which the user can purchase products. Products can be specified at startup and these will be displayed on the Purchase Confirmation screen. The user can purchase these products by pressing the **Confirm Purchase** button. After the products have been purchased, downloading processing will automatically start. Products that are not downloadable such as service entitlements will not be displayed in the list. Also, if the user presses the **Add Funds** button at the upper right corner of Purchase Confirmation screen, the Add Funds screen will be displayed, and the wallet can be funded.

If any of the products specified when the dialog was started up were already purchased or if they cannot be purchased, an error will occur.

Startup in Download Mode

In download mode, the dialog starts up in a state in which the user can only download products. Products can be specified at startup and they will be displayed on the Download List screen. A product can be downloaded by selecting it from the list. Use this mode when providing a function for downloading products that were already purchased.

If any of the products specified when the dialog was started up cannot be downloaded (for example, the products have not yet been purchased), they will not be displayed in the list.

The following methods can be used to specify the product to download.

(1) Specification by SKU ID

If one or more SKU IDs of purchased downloadable products are specified, the products for the specified SKU IDs will be listed on the Download List screen.

(2) Specification by service ID

If a service ID is specified, all purchased and downloadable products with the specified service ID will be listed on the Download List screen.

Startup in Install and Upgrade Mode

In install and upgrade mode, the dialog starts up in a mode in which installation of the downloaded additional contents and upgrade applications to the full version are performed. If there are any additional contents that have already been downloaded, those that are related to applications that are currently running are installed collectively. The user cannot specify individually additional contents to be installed at startup. Similarly, if there are any upgrade licenses that have already been downloaded and those related to the applications that are currently running exist, the upgrade will be performed.

Moreover, unlike other calling modes, in install and upgrade mode, there is no need to call Network Check Dialog and establish a network connection in advance (SDK 2.000 or later).

Funding the Wallet

When the dialog is started up in checkout mode, the user can add funds to the wallet by pressing the [Add Funds] button at the upper right corner of the Purchase Confirmation screen. Either of the following two methods can be selected.

- Credit card
- Prepaid Card for PlayStation®Store

The charge methods that can be selected depend on the country in which the user's account was established.

If the user selects credit card as the funding method and the credit card has not been registered, the Billing Information screen will be displayed and the user can use Store Checkout Dialog to register the credit card.

Display Language

When the Dialog is Started up in Checkout Mode or Download mode

The text in Store Checkout Dialog is displayed in the language of the country in which the user's account was established. Note that the language setting specified to **Language -> System Language** from Settings application is ignored in Store Checkout Dialog.

When the Dialog is Started up in Install and Upgrade Mode

The language setting specified to **Language -> System Language** from Settings application is reflected (SDK 2.100 or later).

4 Precautions

Limitations

- Common Dialog limitations apply.
- Since the screen design may be changed in the future SDK versions, make sure to be independent of the screen design when implementing this library.
- If a directory or file under "addcont0:" is accessed while `sceStoreCheckoutDialogInit()` is being called, the `SCE_STORE_CHECKOUT_DIALOG_ERROR_ADDCONT_BUSY` error results. Call `sceStoreCheckoutDialogInit()` after terminating all accesses to "addcont0:" beforehand. If the additional contents directory is opened using `sceAppUtilDrmOpen()` but no file is accessed, calling `sceStoreCheckoutDialogInit()` is successful, but note that the additional contents opened using `sceAppUtilDrmOpen()` are automatically closed.
- LiveArea™ will be upgraded to its full version during the processing of the upgrade to the full version. In this case, there are precautions regarding the simultaneous use of `libLiveArea`, an update library of LiveArea™. For details, refer to the precautions in the "LiveArea™ Specifications" document.
- Do not use calling modes `SCE_STORE_CHECKOUT_DIALOG_MODE_CHECKOUT` and `SCE_STORE_CHECKOUT_DIALOG_MODE_DOWNLOAD` in SDK 1.600 and later. Instead, use the added calling modes `SCE_STORE_CHECKOUT_DIALOG_MODE_CHECKOUT2` and `SCE_STORE_CHECKOUT_DIALOG_MODE_DOWNLOAD2`.
- Do not use calling mode `SCE_STORE_CHECKOUT_DIALOG_MODE_APPLY` in SDK 1.650 and later. Instead, use the added calling mode `SCE_STORE_CHECKOUT_DIALOG_MODE_INSTALL_AND_UPGRADE`.
- Regarding the SKU ID list specified to `skuIds` of the `SceStoreCheckoutDialogParam` structure to pass to `sceStoreCheckoutDialogInit()`, do not pass hard-coded SKU IDs. It is forbidden for applications to directly hard code SKU IDs. Note that it is possible to pass SKU IDs obtained by using the NP IN-GAME Commerce 2 library.