Store Checkout Dialog Reference

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Table of Contents

Store Checkout Dialog Call

sceStoreCheckoutDialogParamInit

Macro for call parameter initialization

Definition

```
#include <store_checkout_dialog.h>
static inline
void sceStoreCheckoutDialogParamInit(SceStoreCheckoutDialogParam *param)
{
        sceClibMemset( param, 0x0, sizeof(SceStoreCheckoutDialogParam) );
        _sceCommonDialogSetMagicNumber( &param->commonParam );
        param->sdkVersion = SCE_PSP2_SDK_VERSION;
}
```

Calling Conditions

Multithread safe.

Arguments

param	Call parameter

Return Values

None

Description

This is a macro function for initializing the Store Checkout Dialog call parameter.

Before performing the various call parameter settings, be sure to use this macro to execute structure initialization. The appropriate SDK version is set at the same time.

Examples

```
SceStoreCheckoutDialogParam param;

sceStoreCheckoutDialogParamInit( &param );
param.mode = SCE_STORE_CHECKOUT_DIALOG_MODE_CHECKOUT2;
if ( sceStoreCheckoutDialogInit( &param ) < 0 ) {
        // error
}
```

See Also

`SceStoreCheckoutDialogParam`

sceStoreCheckoutDialogInit

Call various functions of Store Checkout Dialog

Definition

```
#include <store_checkout_dialog.h>
SceInt32 sceStoreCheckoutDialogInit(
        const SceStoreCheckoutDialogParam *param
)
```

Calling Conditions

Multithread safe.

Arguments

param	Call parameter

Return Values

Returns `SCE_OK(0)` as the value of the function for success.

Returns one of the following error codes (negative value) for errors.

Value	Hexadecimal	Description
`SCE_COMMON_DIALOG_ERROR_BUSY`	0x80020401	The Common Dialog function is already being called (details below)
`SCE_COMMON_DIALOG_ERROR_NULL`	0x80020402	NULL was specified for the *param* argument
`SCE_COMMON_DIALOG_ERROR_INVALID_INFOBAR_PARAM`	0x80020433	*commonParam.infobarParam* is invalid
`SCE_COMMON_DIALOG_ERROR_INVALID_BG_COLOR`	0x80020434	*commomParam.bgColor* is invalid
`SCE_COMMON_DIALOG_ERROR_INVALID_DIMMER_COLOR`	0x80020435	*commonParam.dimmerColor* is invalid
`SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL`	0x8002047F	Internal error
`SCE_STORE_CHECKOUT_DIALOG_ERROR_PARAM`	0x80102202	Parameter error

Description

This function calls the Store Checkout Dialog function.

This function can be called only when other Common Dialog functions are not called (including the function of Store Checkout Dialog proper). If this function is called at times other than the above, `SCE_COMMON_DIALOG_ERROR_BUSY` is returned.

When calling this function is successful, the operation status immediately changes to `SCE_COMMON_DIALOG_STATUS_RUNNING`. For details on the operation statuses, refer to the `sceStoreCheckoutDialogGetStatus()` section.

In *param*, specify the call parameter structure for which the calling mode was set.

Be sure to set values for *param* after performing initialization with the `sceStoreCheckoutDialogParamInit()` macro.

The *param* instance need not be allocated after this function is called.

Examples

```
SceStoreCheckoutDialogParam param;

sceStoreCheckoutDialogParamInit( &param );
param.mode = SCE_STORE_CHECKOUT_DIALOG_MODE_CHECKOUT2;
if ( sceStoreCheckoutDialogInit( &param ) < 0 ) {
        // error
}
```

See Also

`SceStoreCheckoutDialogParam`, `sceStoreCheckoutDialogParamInit()`, `sceStoreCheckoutDialogGetStatus()`

SceStoreCheckoutDialogParam

Structure for calling Store Checkout Dialog

Definition

```
#include <StoreCheckout_dialog.h>
typedef struct SceStoreCheckoutDialogParam {
        SceUInt32 sdkVersion;
        SceCommonDialogParam commonParam;
        SceStoreCheckoutDialogMode mode;
        SceUInt32 ctxId;
        const SceChar8 **skuIds;
        SceUInt32 skuNum;
        const SceChar8 *serviceId;
        SceChar8 reserved[32];
} SceStoreCheckoutDialogParam;
```

Members

sdkVersion	SDK version
commonParam	Common parameters for Common Dialogs
mode	Calling mode (details below)
ctxId	Context ID
skuIds	List of SKU IDs to be purchased or downloaded
skuNum	The number of SKU IDs included in *skuIds*
serviceId	Service ID to be downloaded
reserved	Reserved area (fill with all 0s)

Description

This is a structure passed to `sceStoreCheckoutDialogInit()` to display Store Checkout Dialog. Use `sceStoreCheckoutDialogParamInit()` to initialize it.

Specify the SDK version in *sdkVersion*. An appropriate value is input when the structure is initialized with `sceStoreCheckoutDialogParamInit()`.

Specify the common parameters for Common Dialogs in *commonParam*.
Given that it is not possible to perform specifications concerning the info bar and background color in Store Checkout Dialog, always specify NULL in *commonParam*.*infobarParam* and *commonParam*.*bgColor*. If values other than NULL are specified, `sceStoreCheckoutDialogInit()` will return `SCE_COMMON_DIALOG_ERROR_INVALID_INFOBAR_PARAM` and `SCE_COMMON_DIALOG_ERROR_INVALID_BG_COLOR`, respectively. For details, refer to the "Common Dialog Reference" document.

Specify the calling mode in `mode`. One of the following values can be input.

Value	Decimal	Description
`SCE_STORE_CHECKOUT_DIALOG_MODE_CHECKOUT2`	2	Starts up in the "checkout mode" for purchasing and downloading Start up in this mode to start the processing from purchasing for the SKU that has not been purchased yet.
`SCE_STORE_CHECKOUT_DIALOG_MODE_DOWNLOAD2`	3	Starts up in the "download mode" for downloading only Start up in this mode to start the processing from downloading for the SKU that has already been purchased.
`SCE_STORE_CHECKOUT_DIALOG_MODE_INSTALL_AND_UPGRADE`	5	Starts up in the "install and upgrade mode" for installing the downloaded additional contents and upgrading applications to the full version.

If a context has been created with the NP IN-GAME Commerce 2 library beforehand, specify the context ID in `ctxId`. If not, specify 0.

Specify the list of SKU IDs to be purchased or downloaded in `skuIds`.

Specify the number of the specified SKU IDs in `skuNum`. Up to 16 SKU IDs can be specified.

Specify the service ID to be downloaded in `serviceId`. This specification is valid when the calling mode is set to `SCE_STORE_CHECKOUT_DIALOG_MODE_DOWNLOAD2`.

In the case of `SCE_STORE_CHECKOUT_DIALOG_MODE_INSTALL_AND_UPGRADE`, do not specify anything in `skuIds`, `skuNum` and `serviceId`.

`reserved` is a reserved area for future function expansion. It must be filled with all 0s.

See Also

`sceStoreCheckoutDialogParamInit()`

Retrieval of Operation Status

sceStoreCheckoutDialogGetStatus

Get operation status of Store Checkout Dialog

Definition

```
#include <store_checkout_dialog.h>
SceCommonDialogStatus sceStoreCheckoutDialogGetStatus(
        void
);
```

Calling Conditions

Multithread safe.

Arguments

None

Return Values

Returns one of the following operation statuses as the value of the function.

Value	Hexadecimal	Description
SCE_COMMON_DIALOG_STATUS_NONE	0x0	Store Checkout Dialog is not running
SCE_COMMON_DIALOG_STATUS_RUNNING	0x1	Store Checkout Dialog is running
SCE_COMMON_DIALOG_STATUS_FINISHED	0x2	Store Checkout Dialog operation has finished

Description

This function gets the operation status of Store Checkout Dialog.

The default value of the operation status is SCE_COMMON_DIALOG_STATUS_NONE.

When calling sceStoreCheckoutDialogInit() is successful, the operation status immediately changes to SCE_COMMON_DIALOG_STATUS_RUNNING.

The operation status of Store Checkout Dialog will not be changed to SCE_COMMON_DIALOG_STATUS_FINISHED until the user taps the close button or sceStoreCheckoutDialogAbort() is called.

sceStoreCheckoutDialogTerm() can be called only while the operation status is SCE_COMMON_DIALOG_STATUS_FINISHED.

When sceStoreCheckoutDialogTerm() is called, the operation status immediately changes to SCE_COMMON_DIALOG_STATUS_NONE.

Examples

```
SceCommonDialogStatus stat = sceStoreCheckoutDialogGetStatus();
```

See Also

sceStoreCheckoutDialogInit(), sceStoreCheckoutDialogAbort(), sceStoreCheckoutDialogGetResult(), sceStoreCheckoutDialogTerm()

Retrieval of Store Checkout Dialog Call Result

sceStoreCheckoutDialogGetResult

Get call result of Store Checkout Dialog

Definition

```
#include <store_checkout_dialog.h>
SceInt32 sceStoreCheckoutDialogGetResult(
        SceStoreCheckoutDialogResult *result
)
```

Calling Conditions

Multithread safe.

Arguments

result Stores the call result

Return Values

Returns SCE_OK(0) as the value of the function for success.

Returns one of the following error codes (negative value) for errors.

Value	Hexadecimal	Description
SCE_COMMON_DIALOG_ERROR_NULL	0x80020402	NULL was specified for the *result* argument
SCE_COMMON_DIALOG_ERROR_NOT_FINISHED	0x80020410	Called during other than the appropriate operation status (details below)
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error
SCE_STORE_CHECKOUT_DIALOG_ERROR_PARAM	0x80102202	Parameter error

Description

This function retrieves the call result of Store Checkout Dialog.

This function can be called only while the operation status of Store Checkout Dialog is SCE_COMMON_DIALOG_STATUS_FINISHED. If it is called at times other than the above, SCE_COMMON_DIALOG_ERROR_NOT_FINISHED is returned. For details on the operation statuses, refer to the sceStoreCheckoutDialogGetStatus() section.

The call result of Store Checkout Dialog is stored in *result*. For details on the call results, refer to the SceStoreCheckoutDialogResult section.

Note that if the return value of the function is not SCE_OK, *result* will be undefined.

Be sure to initialize the argument *result* before passing it to this function. If SceStoreCheckoutDialogResult.*reserved* is not filled with all 0s, SCE_STORE_CHECKOUT_DIALOG_ERROR_PARAM will be returned as the value of the function.

Examples

```
SceCommonDialogStatus stat;

while(1) {
        stat = sceStoreCheckoutDialogGetStatus();
        if( stat == SCE_COMMON_DIALOG_STATUS_FINISHED ) {
              SceStoreCheckoutDialogResult result;
              sceClibMemset( &result, 0, sizeof(result) );
              if( 0 > sceStoreCheckoutDialogGetResult( &result ) ) {
                     // error
              }
              sceStoreCheckoutDialogTerm();
              break;
        }
}
```

See Also

`SceStoreCheckoutDialogResult,sceStoreCheckoutDialogGetStatus()`

SceStoreCheckoutDialogResult

Structure for retrieving Store Checkout Dialog call result

Definition

```
#include <store_checkout_dialog.h>
typedef struct SceStoreCheckoutDialogResult {
        SceInt32 result;
        SceInt32 npError;
        SceChar8 reserved[32];
} SceStoreCheckoutDialogResult;
```

Members

result	Stores the call result (details below)
npError	NP library error code
reserved	Reserved area (fill with all 0s)

Description

This structure receives the Store Checkout Dialog call result. It is passed to `sceStoreCheckoutDialogGetResult()`. Fill it with all 0s during initialization.

The call result of Store Checkout Dialog is stored in *result*. In the case of success, one of the following positive values is stored.

Value	Hexadecimal	Description
SCE_COMMON_DIALOG_RESULT_OK	0x0	Processing completed
SCE_COMMON_DIALOG_RESULT_USER_CANCELED	0x1	User performed cancel operation.
SCE_COMMON_DIALOG_RESULT_ABORTED	0x2	Abortion through `sceStoreCheckoutDialogAbort()`

In the case of an error, one of the following error codes (negative value) is stored.

Value	Hexadecimal	Description
SCE_STORE_CHECKOUT_DIALOG_ERROR_INTERNAL	0x80102201	Internal error
SCE_STORE_CHECKOUT_DIALOG_ERROR_PARAM	0x80102202	Parameter error
SCE_STORE_CHECKOUT_DIALOG_ERROR_NP	0x80102203	NP library error (An error code is stored in *npError*)
SCE_STORE_CHECKOUT_DIALOG_ERROR_ADDCONT_BUSY	0x80102204	Additional contents currently in use

The error of the NP IN-GAME Commerce 2 library will be returned to *npError*. An error code will be stored in *npError* when an error occurs in the NP IN-GAME Commerce 2 library during purchasing or downloading, resulting in a failure of the purchase or download processing.

reserved is a reserved area for future function expansion. It must be filled with all 0s.

See Also

`sceStoreCheckoutDialogInit()`, `sceStoreCheckoutDialogAbort()`, `sceStoreCheckoutDialogGetResult()`, `SceStoreCheckoutDialogParam`

Abortion of Store Checkout Dialog

sceStoreCheckoutDialogAbort

Abort call of Store Checkout Dialog

Definition

```
#include <store_checkout_dialog.h>
SceInt32 sceStoreCheckoutDialogAbort(
        void
);
```

Calling Conditions

Multithread safe.

Arguments

None

Return Values

Returns SCE_OK(0) as the value of the function for success.

Returns one of the following error codes (negative value) for errors.

Value	Hexadecimal	Description
SCE_COMMON_DIALOG_ERROR_NOT_IN_USE	0x80020411	sceStoreCheckoutDialogInit() has not been called
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

Description

This function aborts calling of Store Checkout Dialog.

It can be called at any time between when sceStoreCheckoutDialogInit() is called and sceStoreCheckoutDialogTerm() is called.
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.
For details on the operation statuses, refer to the sceStoreCheckoutDialogGetStatus() section.

When Store Checkout Dialog is closed with this function, calling sceStoreCheckoutDialogGetResult() returns the following.

SceStoreCheckoutDialogResult.*result* : SCE_COMMON_DIALOG_RESULT_ABORTED

sceStoreCheckoutDialogAbort() is used to promptly abort the Store Checkout Dialog display, for example when an urgent interrupt must be processed.

Examples

```
SceCommonDialogStatus stat;

while(1) {
        stat = sceStoreCheckoutDialogGetStatus();
        if( stat == SCE_COMMON_DIALOG_STATUS_RUNNING ) {
              if( need_abort ) {
                     sceStoreCheckoutDialogAbort();
                     break;
              }
        }
        else if( stat == SCE_COMMON_DIALOG_STATUS_FINISHED ) {
              sceStoreCheckoutDialogTerm();
              break;
        }
}
```

See Also

`sceStoreCheckoutDialogGetStatus(),sceStoreCheckoutDialogGetResult()`

End Store Checkout Dialog

sceStoreCheckoutDialogTerm

End call of Store Checkout Dialog

Definition

```
#include <store_checkout_dialog.h>
SceInt32 sceStoreCheckoutDialogTerm(
        void
);
```

Calling Conditions

Multithread safe.

Arguments

None

Return Values

Returns `SCE_OK(0)` as the value of the function for success.

Returns one of the following error codes (negative value) for errors.

Value	Hexadecimal	Description
`SCE_COMMON_DIALOG_ERROR_NOT_FINISHED`	0x80020410	Called during other than the appropriate operation status (details below)
`SCE_COMMON_DIALOG_ERROR_NOT_IN_USE`	0x80020411	`sceStoreCheckoutDialogInit()` has not been called
`SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL`	0x8002047F	Internal error

Description

This function ends calling of Store Checkout Dialog. Store Checkout Dialog must be ended with this function after it has been called with `sceStoreCheckoutDialogInit()`.

This function can be called only while the operation status of Store Checkout Dialog is `SCE_COMMON_DIALOG_STATUS_FINISHED`. If it is called during times other than the above, `SCE_COMMON_DIALOG_ERROR_NOT_FINISHED` is returned. `SCE_COMMON_DIALOG_ERROR_NOT_IN_USE` will be returned if the Store Checkout Dialog function is not called.

If calling this function is successful, the operation status changes immediately to `SCE_COMMON_DIALOG_STATUS_NONE`. For details on the operation statuses, refer to the `sceStoreCheckoutDialogGetStatus()` section.

Examples

```
SceCommonDialogStatus stat;

while(1) {
        stat = sceStoreCheckoutDialogGetStatus();
        if( stat == SCE_COMMON_DIALOG_STATUS_FINISHED ) {
              sceStoreCheckoutDialogTerm();
              break;
        }
}
```

See Also

`sceStoreCheckoutDialogInit(),sceStoreCheckoutDialogGetStatus()`

Constants

Return Codes

List of return codes returned by Store Checkout Dialog

Definition

Value	Hexadecimal	Description
SCE_COMMON_DIALOG_ERROR_BUSY	0x80020401	Calling another common dialog function
SCE_COMMON_DIALOG_ERROR_NULL	0x80020402	NULL was specified as the function's argument
SCE_COMMON_DIALOG_ERROR_INVALID_ARGUMENT	0x80020403	Parameter error
SCE_COMMON_DIALOG_ERROR_NOT_RUNNING	0x80020404	Called during a period other than SCE_COMMON_DIALOG_STATUS_RUNNING
SCE_COMMON_DIALOG_ERROR_NOT_FINISHED	0x80020410	Called during a period other than SCE_COMMON_DIALOG_STATUS_FINISHED
SCE_COMMON_DIALOG_ERROR_NOT_IN_USE	0x80020411	sceStoreCheckoutDialogInit() has not been called
SCE_COMMON_DIALOG_ERROR_INVALID_INFOBAR_PARAM	0x80020433	There is an invalid setting in *commonParam.infobarParam*
SCE_COMMON_DIALOG_ERROR_INVALID_BG_COLOR	0x80020434	There is an invalid setting in *commonParam.bgColor*
SCE_COMMON_DIALOG_ERROR_INVALID_DIMMER_COLOR	0x80020435	There is an invalid setting in *commonParam.dimmerColor*
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error
SCE_STORE_CHECKOUT_DIALOG_ERROR_INTERNAL	0x80102201	Internal error
SCE_STORE_CHECKOUT_DIALOG_ERROR_PARAM	0x80102202	Parameter error
SCE_STORE_CHECKOUT_DIALOG_ERROR_NP	0x80102203	NP library error
SCE_STORE_CHECKOUT_DIALOG_ERROR_ADDCONT_BUSY	0x80102204	Additional contents currently in use

Parameter Errors

Codes output when values of structures specified to Store Checkout Dialog are invalid

Definition

Value	Description
1	`SceStoreCheckoutDialogParam` is NULL
2	`SceStoreCheckoutDialogParam.`*`reserved`* is invalid
3	`SceStoreCheckoutDialogParam.`*`mode`* is invalid
100	`SceStoreCheckoutDialogResult` is NULL
101	`SceStoreCheckoutDialogResult.`*`reserved`* is invalid

Description

If the contents of the structures specified for the various functions provided by Store Checkout Dialog are invalid, they are processed as parameter errors, and the operation status of Store Checkout Dialog changes immediately to `SCE_COMMON_DIALOG_STATUS_FINISHED`.

At this time, `SCE_STORE_CHECKOUT_DIALOG_ERROR_PARAM` is stored in `SceStoreCheckoutDialogResult.`*`result`*, which can be retrieved with `sceStoreCheckoutDialogGetResult()`.

Furthermore, the concrete parameter error occurrence locations are output to the console in the following format.

```
****** SceStoreCheckoutDialog Parameter Error : XX ******
```

(`XX` is one of the above numbers)

If, exceptionally, a parameter error occurs for `sceStoreCheckoutDialogGetResult()`, `SCE_STORE_CHECKOUT_DIALOG_ERROR_PARAM` is returned as the return value of the API.

Parameter errors are coding mistakes of the application and must absolutely be fixed before release.