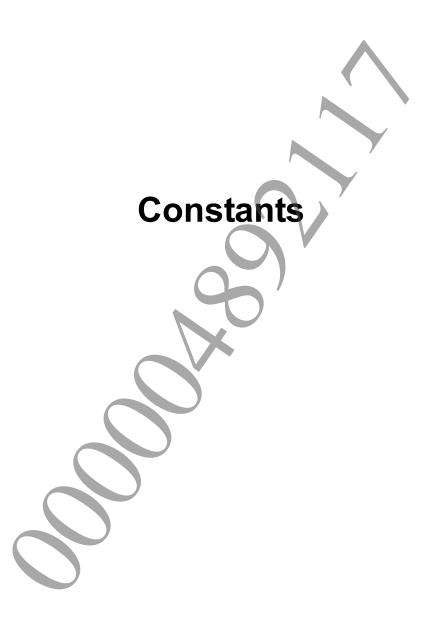


© 2013 Sony Computer Entertainment Inc. All Rights Reserved. SCE Confidential

## **Table of Contents**

Constants	3
SceCommonDialogStatus	4
SceCommonDialogResult	
Datatypes	6
SceCommonDialogRenderTargetInfo	
SceCommonDialogUpdateParam	
SceCommonDialogInfobarParam	
SceCommonDialogColor	
SceCommonDialogParam	11
Functions	
sceCommonDialogGetWorkerThreadId	13
sceCommonDialogIsRunning	14
sceCommonDialogUpdate	
Error Codes	
Error Codes	



# SceCommonDialogStatus

## State of Common Dialog

### **Definition**

#include <common\_dialog/types.h>
typedef enum SceCommonDialogStatus;

## **Enumeration Values**

Value	(Number)	Description
SCE_COMMON_DIALOG_STATUS_NONE	0	Common Dialog is not initialized
SCE_COMMON_DIALOG_STATUS_RUNNING	1	Common Dialog is running
SCE COMMON DIALOG STATUS FINISHED	2	Common Dialog has been finished

## **Description**

These constants define the state of Common Dialog.

### See Also

sceMsgDialogGetStatus(), sceSaveDataDialogGetStatus(),
sceNetCheckDialogGetStatus()



# SceCommonDialogResult

## **Execution result of Common Dialog**

## **Definition**

#include <common\_dialog/types.h>
typedef enum SceCommonDialogResult;

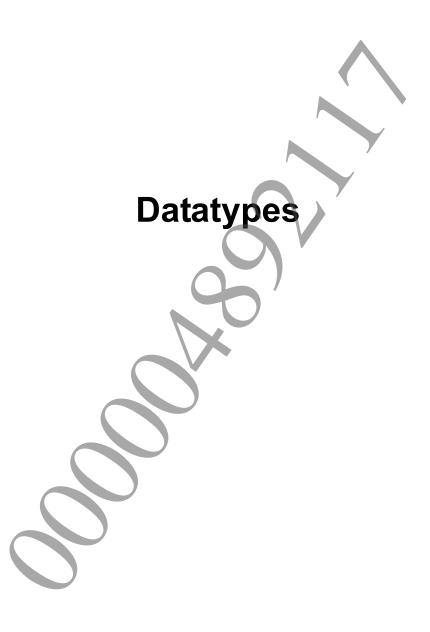
## **Enumeration Values**

Value	(Number)	Description
SCE_COMMON_DIALOG_RESULT_OK	0	Execution has been finished normally
SCE_COMMON_DIALOG_RESULT_USER_CANCELED	1	Execution has been canceled by the user
SCE_COMMON_DIALOG_RESULT_ABORTED	2	Execution has been aborted

### **Description**

These constants define the execution result of Common Dialog.





## SceCommonDialogRenderTargetInfo

Information on rendering target of Common Dialog

#### **Definition**

```
#include <common_dialog/types.h>
typedef struct SceCommonDialogRenderTargetInfo {
    ScePVoid colorSurfaceData;
    SceGxmColorSurfaceType surfaceType;
    SceGxmColorFormat colorFormat;
    ScePVoid depthSurfaceData;
    SceUInt32 width;
    SceUInt32 height;
    SceUInt32 strideInPixels;
    SceUInt8 reserved[];
} SceCommonDialogRenderTargetInfo;
```

#### **Members**

colorSurfaceData Pointer to surface data of color surface
surfaceType Memory layout of color surface
colorFormat Color format of color surface

depthSurfaceDataPointer to surface data of depth surfacewidthWidth of color surface and depth surfaceheightHeight of color surface and depth surface

strideInPixels Stride of color surface and depth surface (in pixel unit)
reserved Reserved area (Make sure to zero-fill the area)

Description

Common Dialog is directly rendered on the color surface of the application. Therefore, the application must pass the various parameters of its color surface to Common Dialog through this structure.

In this structure, it is also possible to write depth surface parameters. However, in this release of the SDK there are no dialogs that actually require a depth buffer. It is possible to specify a NULL pointer in depthSurfaceData.

### See Also

SceCommonDialogUpdateParam, sceCommonDialogUpdate()

**©SCEI** 

## SceCommonDialogUpdateParam

## Update parameter of Common Dialog

### **Definition**

```
#include <common dialog/types.h>
typedef struct SceCommonDialogUpdateParam {
        SceCommonDialogRenderTargetInfo renderTarget;
        SceGxmSyncObject *displaySyncObject;
        SceUInt8 reserved[];
} SceCommonDialogUpdateParam;
```

#### **Members**

renderTarget Information on rendering target of Common Dialog

Synchronization object that performs notification upon completing the displaySyncObject

rendering of Common Dialog

reserved Reserved area (Make sure to zero the area)

### **Description**

This structure is used to store the various parameters which are required to update Common Dialog.

## See Also

SceCommonDialogRenderTargetInfo, sceCommonDialogUpdate()



## SceCommonDialogInfobarParam

### Info bar parameters

### **Definition**

#### **Members**

visibility Presence of info bar display

color Info bar color

transparency Info bar transparency

reserved Reserved area (Make sure to zero the area)

### **Description**

This structure is used for controlling info bar status while Common Dialog is running. Specify one of the following macros in <code>visibility</code>.

Value	Description
SCE_APPMGR_INFOBAR_VISIBILITY_VISIBLE	Displays info bar
SCE APPMGR INFOBAR VISIBILITY INVISIBLE	Does not display info bar

When having the info bar displayed, it is possible to select color and transparency with color and transparency, respectively. Specify one of the following macros in color.

Value		Description
		Makes the info bar black.
SCE APPMGR INFOBAR	COLOR WHITE	Makes the info bar white.

Specify one of the following macros in transparency.

Value		Description
SCE_APPMGR_INFOBAR	TRANSPARENCY_OPAQUE	Makes the info bar opaque.
SCE APPMGR INFOBAR	TRANSPARENCY TRANSLUCENT	Makes the info bar translucent.

#### See Also

SceCommonDialogParam

## SceCommonDialogColor

## Color of Common Dialog

### **Definition**

### **Members**

- r Red component (0 255)
- g Green component (0 255)
- b Blue component (0 255)
- a Alpha component (0 255)

### **Description**

Use this structure to control the color of Common Dialog.

### See Also

SceCommonDialogParam



## SceCommonDialogParam

## Common parameters for Common Dialogs

#### **Definition**

#### **Members**

infobarParam Info bar parameters

bgColor Background color of Common Dialog

dimmerColor Dimmer color

reserved Reserved area (Make sure to zero the area)

magic Magic number

### **Description**

Use this structure to control each type of common parameter for Common Dialogs.

Specify <code>infobarParam</code>, <code>bgColor</code> and <code>dimmerColor</code> by using the pointer to their respective structures. At the moment in which <code>SceCommonDialogParam</code> is initialized, the value of each pointer is <code>NULL</code>. Leaving them as they are means that default parameters will be specified.

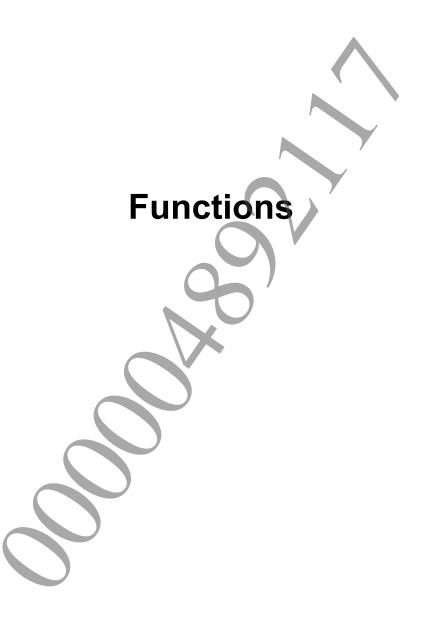
magic is a value automatically set when a dialog is initialized.

Default values of each parameter vary according to the type of dialog, and in some cases it may not be possible to modify default settings.

#### See Also

SceCommonDialogInfobarParam, SceCommonDialogColor





## sceCommonDialogGetWorkerThreadId

Get worker thread identifier (thread ID) used by Common Dialog

### **Definition**

#include <common dialog/common api.h> SceUID sceCommonDialogGetWorkerThreadId();

### **Arguments**

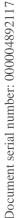
None

#### **Return Values**

Value	Description
Positive value	Worker thread identifier (thread ID)
Negative value	Error code

### **Description**

This function obtains the worker thread identifier (thread ID) used by Common Dialog. The priority and CPU affinity mask of the worker thread can be changed. The worker thread does not always exist when Common Dialog is not working. Therefore, after initializing each Common Dialog, obtain the worker thread identifier then change the priority and CPU affinity mask. If a worker thread does not exist, SCE UID INVALID UID (-1) will return.



# sceCommonDialogIsRunning

Get information on whether or not Common Dialog is running

### **Definition**

#include <common\_dialog/common\_api.h>
SceBool sceCommonDialogIsRunning();

## **Arguments**

None

### **Return Values**

Value	Description
SCE_TRUE	Common Dialog is running
SCE_FALSE	Common Dialog is not running

## **Description**

This function returns whether or not one of the dialogs is in a SCE COMMON DIALOG STATUS RUNNING state.



## sceCommonDialogUpdate

### **Update Common Dialog**

### **Definition**

### **Arguments**

param Required parameter for updating Common Dialog

### **Return Values**

Returns SCE\_OK(0) for success.

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

Value	(Number)	Description
SCE_COMMON_DIALOG_ERROR_	0x80020402	NULL was specified in param
NULL		NULL was specified in
		param->renderTarget.colorSurfaceData
SCE_COMMON_DIALOG_ERROR_	0x80020406	This function was called in libgxm display-queue
ILLEGAL_CALLER_THREAD		callback.
SCE_COMMON_DIALOG_ERROR_	0x80020420	param->renderTarget.colorFormat is not
INVALID_COLOR_FORMAT		SCE_GXM_COLOR_FORMAT_U8U8U8U8_ABGR
SCE_COMMON_DIALOG_ERROR_	0x80020421	Invalid value was specified in
INVALID_SURFACE_RESOLUTION		param->renderTarget.widthorheight
SCE_COMMON_DIALOG_ERROR_	0x80020422	Invalid value was specified in
INVALID_SURFACE_STRIDE		param->renderTarget.strideInPixels
SCE_COMMON_DIALOG_ERROR_	0x80020423	param->renderTarget.surfaceType is not
INVALID_SURFACE_TYPE		SCE_GXM_COLOR_SURFACE_LINEAR
SCE_COMMON_DIALOG_ERROR_	0x80020424	Called within a scene of the application
WITHIN_SCENE		(sceGxmEndScene() has not been called)

### **Description**

This function updates Common Dialog according to the parameter specified in the argument param and must be called by each rendering frame during the use of Common Dialog.

The rendering of Common Dialog performed by GPU is not completed at the time of returning from the function. In order to flip the frame buffer after the rendering of Common Dialog is completed, synchronization must be done by using the synchronization object set in <code>param->displaySyncObject</code>.

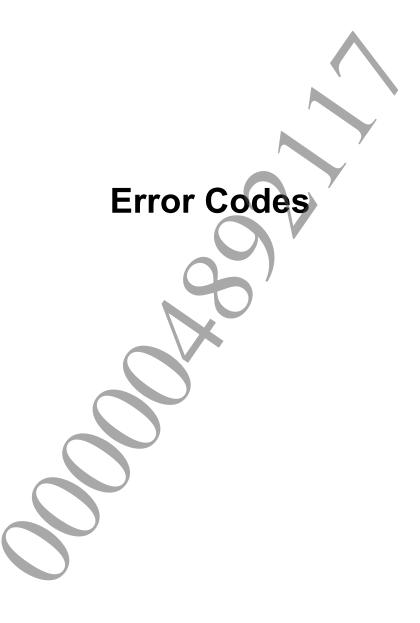
### **Notes**

This function immediately returns SCE\_OK without executing any processing if it is called when Common Dialog is not in use.

### See Also

SceCommonDialogUpdateParam

**©SCEI** 



## **Error Codes**

## List of error codes

## Definition

Value	(Number)	Description
		-
SCE_COMMON_DIALOG_ERROR_BUSY	0x80020401	Another Common Dialog is in use
SCE_COMMON_DIALOG_ERROR_NULL	0x80020402	Invalid NULL pointer has been specified in
		argument
SCE_COMMON_DIALOG_ERROR_	0x80020403	Parameter error
INVALID_ARGUMENT		
SCE_COMMON_DIALOG_ERROR_NOT_	0x80020404	Called during a period other than
RUNNING		SCE COMMON DIALOG STATUS RUNNING
SCE_COMMON_DIALOG_ERROR_NOT_	0x80020405	Called in an operation mode not supported by
SUPPORTED		the API
SCE_COMMON_DIALOG_ERROR_	0x80020406	The API was called by an illegal thread.
ILLEGAL_CALLER_THREAD		
SCE_COMMON_DIALOG_ERROR_NOT_	0x80020407	Initialization failed due to a Common Dialog
CONFIGURED		internal error
SCE_COMMON_DIALOG_ERROR_NOT_	0x80020410	Attempted to finish Common Dialog during
FINISHED		the execution
SCE_COMMON_DIALOG_ERROR_NOT_	0x80020411	The Common Dialog concerned is not
IN_USE		currently in use
SCE_COMMON_DIALOG_ERROR_	0x80020420	Invalid color format was specified
INVALID COLOR FORMAT		
SCE COMMON DIALOG ERROR	0x80020421	Invalid surface resolution is specified
INVALID_SURFACE_RESOLUTION		
SCE_COMMON_DIALOG_ERROR_	0x80020422	Invalid surface stride is specified
INVALID_SURFACE_STRIDE		•
SCE_COMMON_DIALOG_ERROR_	0x80020423	Invalid surface type is specified
INVALID_SURFACE_TYPE		
SCE_COMMON_DIALOG_ERROR_	0x80020424	Tried to render Common Dialog inside a scene
WITHIN_SCENE		of the application
SCE_COMMON_DIALOG_ERROR	0x80020430	Cannot open Common Dialog because libime
IME_IN_USE		is being used
SCE COMMON DIALOG ERROR	0x80020431	Invalid language was specified
INVALID LANGUAGE		0 0 1
SCE_COMMON_DIALOG ERROR	0x80020432	Invalid Enter button was specified
INVALID ENTER BUTTON ASSIGN		1
SCE_COMMON_DIALOG_ERROR_	0x80020433	Invalid info bar parameter was specified
INVALID_INFOBAR_PARAM		•
SCE_COMMON_DIALOG_ERROR_	0x80020434	Invalid background color was specified
INVALID_BG_COLOR		•
SCE_COMMON_DIALOG_ERROR_	0x80020435	Invalid dimmer color was specified
INVALID_DIMMER_COLOR		
SCE_COMMON_DIALOG_ERROR_GXM_	0x80020436	libgxm is not initialized
IS_UNINITIALIZED		
SCE_COMMON_DIALOG_ERROR_	0x8002047F	Unexpected fatal error occurred
UNEXPECTED_FATAL		

## **Description**

Each API of Common Dialog returns  $SCE_OK(0)$  for success or the above-described return codes (negative value) for errors. Uniquely defined error codes may be returned depending on the type of the dialog.

**©SCEI**