

# Common Dialog Reference

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

## Table of Contents

---

<b>Constants .....</b>	<b>3</b>
SceCommonDialogStatus .....	4
SceCommonDialogResult .....	5
<b>Datatypes .....</b>	<b>6</b>
SceCommonDialogRenderTargetInfo .....	7
SceCommonDialogUpdateParam .....	8
SceCommonDialogInfobarParam .....	9
SceCommonDialogColor .....	10
SceCommonDialogParam .....	11
<b>Functions .....</b>	<b>12</b>
sceCommonDialogGetWorkerThreadId .....	13
sceCommonDialogIsRunning .....	14
sceCommonDialogUpdate .....	15
<b>Error Codes .....</b>	<b>16</b>
Error Codes .....	17

## Constants

000004892117

SCE CONFIDENTIAL

---

# 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()
```

SCE CONFIDENTIAL

---

# 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.

# Datatypes

000004892117

SCE CONFIDENTIAL

# 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

<i>colorSurfaceData</i>	Pointer to surface data of color surface
<i>surfaceType</i>	Memory layout of color surface
<i>colorFormat</i>	Color format of color surface
<i>depthSurfaceData</i>	Pointer to surface data of depth surface
<i>width</i>	Width of color surface and depth surface
<i>height</i>	Height of color surface and depth surface
<i>strideInPixels</i>	Stride of color surface and depth surface (in pixel unit)
<i>reserved</i>	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()

SCE CONFIDENTIAL

# SceCommonDialogUpdateParam

Update parameter of Common Dialog

## Definition

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

## Members

<i>renderTarget</i>	Information on rendering target of Common Dialog
<i>displaySyncObject</i>	Synchronization object that performs notification upon completing the rendering of Common Dialog
<i>reserved</i>	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()`



SCE CONFIDENTIAL

# SceCommonDialogInfobarParam

## Info bar parameters

### Definition

```
#include <common_dialog/types.h>
typedef struct SceCommonDialogInfobarParam {
    SceInt32 visibility;
    SceInt32 color;
    SceInt32 transparency;
    SceUInt8 reserved[];
} SceCommonDialogInfobarParam;
```

### 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 *visibility*.

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
SCE_APPMGR_INFOBAR_COLOR_BLACK	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

SCE CONFIDENTIAL

---

# SceCommonDialogColor

---

## Color of Common Dialog

### Definition

---

```
#include <common_dialog/types.h>
typedef struct SceCommonDialogColor {
    SceUInt8 r;
    SceUInt8 g;
    SceUInt8 b;
    SceUInt8 a;
} SceCommonDialogColor;
```

### Members

---

<i>r</i>	Red component (0 - 255)
<i>g</i>	Green component (0 - 255)
<i>b</i>	Blue component (0 - 255)
<i>a</i>	Alpha component (0 - 255)

### Description

---

Use this structure to control the color of Common Dialog.

### See Also

---

SceCommonDialogParam

SCE CONFIDENTIAL

# SceCommonDialogParam

## Common parameters for Common Dialogs

### Definition

```
#include <common_dialog/types.h>
typedef struct SceCommonDialogParam {
    SceCommonDialogInfoBarParam* infoBarParam;
    SceCommonDialogColor* bgColor;
    SceCommonDialogColor* dimmerColor;
    SceUInt8 reserved[];
    SceUInt32 magic;
} SceCommonDialogParam;
```

### Members

<i>infoBarParam</i>	Info bar parameters
<i>bgColor</i>	Background color of Common Dialog
<i>dimmerColor</i>	Dimmer color
<i>reserved</i>	Reserved area (Make sure to zero the area)
<i>magic</i>	Magic number

### Description

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

Specify *infoBarParam*, *bgColor* and *dimmerColor* by using the pointer to their respective structures. At the moment in which *SceCommonDialogParam* is initialized, the value of each pointer is NULL. 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*

# Functions

000004892117

SCE CONFIDENTIAL

---

## 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.

SCE CONFIDENTIAL

---

## 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.

SCE CONFIDENTIAL

# sceCommonDialogUpdate

## Update Common Dialog

### Definition

```
#include <common_dialog/common_api.h>
SceInt32 sceCommonDialogUpdate(
    const SceCommonDialogUpdateParam *param
);
```

### 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_NULL	0x80020402	NULL was specified in <i>param</i> NULL was specified in <i>param-&gt;renderTarget.colorSurfaceData</i>
SCE_COMMON_DIALOG_ERROR_ILLEGAL_CALLER_THREAD	0x80020406	This function was called in libgxm display-queue callback.
SCE_COMMON_DIALOG_ERROR_INVALID_COLOR_FORMAT	0x80020420	<i>param-&gt;renderTarget.colorFormat</i> is not SCE_GXM_COLOR_FORMAT_U8U8U8U8_ABGR
SCE_COMMON_DIALOG_ERROR_INVALID_SURFACE_RESOLUTION	0x80020421	Invalid value was specified in <i>param-&gt;renderTarget.width</i> or <i>height</i>
SCE_COMMON_DIALOG_ERROR_INVALID_SURFACE_STRIDE	0x80020422	Invalid value was specified in <i>param-&gt;renderTarget.strideInPixels</i>
SCE_COMMON_DIALOG_ERROR_INVALID_SURFACE_TYPE	0x80020423	<i>param-&gt;renderTarget.surfaceType</i> is not SCE_GXM_COLOR_SURFACE_LINEAR
SCE_COMMON_DIALOG_ERROR_WITHIN_SCENE	0x80020424	Called within a scene of the application ( <i>sceGxmEndScene()</i> 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 *param->displaySyncObject*.

### 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

## Error Codes

000004892117



SCE CONFIDENTIAL

# 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_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_SUPPORTED	0x80020405	Called in an operation mode not supported by the API
SCE_COMMON_DIALOG_ERROR_ILLEGAL_CALLER_THREAD	0x80020406	The API was called by an illegal thread.
SCE_COMMON_DIALOG_ERROR_NOT_CONFIGURED	0x80020407	Initialization failed due to a Common Dialog internal error
SCE_COMMON_DIALOG_ERROR_NOT_FINISHED	0x80020410	Attempted to finish Common Dialog during the execution
SCE_COMMON_DIALOG_ERROR_NOT_IN_USE	0x80020411	The Common Dialog concerned is not currently in use
SCE_COMMON_DIALOG_ERROR_INVALID_COLOR_FORMAT	0x80020420	Invalid color format was specified
SCE_COMMON_DIALOG_ERROR_INVALID_SURFACE_RESOLUTION	0x80020421	Invalid surface resolution is specified
SCE_COMMON_DIALOG_ERROR_INVALID_SURFACE_STRIDE	0x80020422	Invalid surface stride is specified
SCE_COMMON_DIALOG_ERROR_INVALID_SURFACE_TYPE	0x80020423	Invalid surface type is specified
SCE_COMMON_DIALOG_ERROR_WITHIN_SCENE	0x80020424	Tried to render Common Dialog inside a scene of the application
SCE_COMMON_DIALOG_ERROR_IME_IN_USE	0x80020430	Cannot open Common Dialog because libime is being used
SCE_COMMON_DIALOG_ERROR_INVALID_LANGUAGE	0x80020431	Invalid language was specified
SCE_COMMON_DIALOG_ERROR_INVALID_ENTER_BUTTON_ASSIGN	0x80020432	Invalid Enter button was specified
SCE_COMMON_DIALOG_ERROR_INVALID_INFOBAR_PARAM	0x80020433	Invalid info bar parameter was specified
SCE_COMMON_DIALOG_ERROR_INVALID_BG_COLOR	0x80020434	Invalid background color was specified
SCE_COMMON_DIALOG_ERROR_INVALID_DIMMER_COLOR	0x80020435	Invalid dimmer color was specified
SCE_COMMON_DIALOG_ERROR_GXM_IS_UNINITIALIZED	0x80020436	libgxm is not initialized
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Unexpected fatal error occurred

### 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.