

# Cross-Controller Dialog Reference

© 2012 Sony Computer Entertainment Inc.  
All Rights Reserved.  
SCE Confidential

## Table of Contents

<b>Cross-Controller Dialog Call .....</b>	<b>3</b>
sceCrossControllerDialogParamInit .....	4
sceCrossControllerDialogInit .....	5
SceCrossControllerDialogParam .....	7
<b>Obtainment of Operation Status .....</b>	<b>8</b>
sceCrossControllerDialogGetStatus .....	9
<b>Obtainment of Cross-Controller Dialog Call Result .....</b>	<b>10</b>
sceCrossControllerDialogGetResult .....	11
SceCrossControllerDialogResult .....	13
<b>Aborting Cross-Controller Dialog.....</b>	<b>14</b>
sceCrossControllerDialogAbort.....	15
<b>Termination of Cross-Controller Dialog.....</b>	<b>17</b>
sceCrossControllerDialogTerm .....	18
<b>Constants .....</b>	<b>19</b>
Return Codes .....	20

# Cross-Controller Dialog Call

SCE CONFIDENTIAL

# sceCrossControllerDialogParamInit

Macro function for call parameter initialization

## Definition

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

## Arguments

*param* Call parameter

## Return Values

None

## Description

This is a macro function for initializing the Cross-Controller Dialog call parameters.

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

```
SceCrossControllerDialogParam param;
sceCrossControllerDialogParamInit (&param);

strncpy(param.ps3TitleId, "ABCD12345", sizeof(param.ps3TitleId));
strncpy(param.ps3AppMinVer, "1.00", sizeof(param.ps3AppMinVer));
if (sceCrossControllerDialogInit( &param ) < 0 ) {
    // error
}
```

## See Also

SceCrossControllerDialogParam

SCE CONFIDENTIAL

# sceCrossControllerDialogInit

## Call Cross-Controller Dialog

### Definition

```
#include <cross_controller_dialog.h>
SceInt32
sceCrossControllerDialogInit(
    const SceCrossControllerDialogParam *initParam
)
```

### Arguments

*initParam* 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 argument <i>initParam</i>
SCE_COMMON_DIALOG_ERROR_INVALID_ARGUMENT	0x80020403	Invalid value was specified for the argument <i>initParam</i>
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

### Description

This function calls Cross-Controller Dialog

This function can be called only when other Common Dialog functions are not called (including the Cross-Controller Dialog function 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 sceCrossControllerDialogGetStatus() section.

In *initParam*, specify the call parameter structure for which the operation mode was set.

Be sure to set the various values after performing initialization of *initParam* with the sceCrossControllerDialogParamInit() macro function first.

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

This function is multithread safe.

SCE CONFIDENTIAL

---

## Examples

---

```
SceCrossControllerDialogParam param;

sceCrossControllerDialogParamInit( &param );
strncpy(param.ps3TitleId, "ABCD12345", sizeof(param.ps3TitleId));
strncpy(param.ps3AppMinVer, "1.00", sizeof(param.ps3AppMinVer));
if ( sceCrossControllerDialogInit( &param ) < 0 ) {
    // error
}
```

## See Also

---

SceCrossControllerDialogParam, sceCrossControllerDialogParamInit(),  
sceCrossControllerDialogGetStatus()

SCE CONFIDENTIAL

# SceCrossControllerDialogParam

## Cross-Controller Dialog parameter structure

### Definition

```
#include <cross_controller_dialog.h>
typedef struct SceCrossControllerDialogParam {
    SceUInt32 sdkVersion;
    SceCommonDialogParam commonParam;
    char ps3TitleId[9+1];
    char ps3AppMinVer[5+1];
    SceBool isNoCancel;
    SceUInt8 reserved[12];
} SceCrossControllerDialogParam;
```

### Members

<i>sdkVersion</i>	SDK version
<i>commonParam</i>	Common parameters for Common Dialogs
<i>ps3TitleId</i>	ID of the title expected to be running on PlayStation®3
<i>ps3AppMinVer</i>	Version of the title expected to be running on PlayStation®3
<i>isNoCancel</i>	The Cancel button will not display
<i>reserved</i>	Reserved area

### Description

This is a structure passed to `sceCrossControllerDialogInit()` to display Cross-Controller Dialog. Use `sceCrossControllerDialogParamInit()` to initialize it.

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

Specify the common parameters for Common Dialogs in *commonParam*.

Specify background color information in ARGB format (0 - 255) in *commonParam.bgColor*. For use in the game, only 0 (transparent) or 255 (non-transparent) can be specified in *commonParam.bgColor.a*.

Specify dimmer color information in ARGB format (0 - 255) in *commonParam.dimmerColor*. Only (r,g,b,a)=(0,0,0,0) (transparent) or (r,g,b,a)=(0,0,0,255) (non-transparent black) can be used.

For details on the common parameters for Common Dialogs, refer to the "Common Dialog Reference" document.

Specify the ID of the title expected to be running on PlayStation®3 in *ps3TitleId*.

Specify the minimum version of the title expected to be running on PlayStation®3 in *ps3AppMinVer*.

When *isNoCancel* is set to `SCE_TRUE`, the Cancel button will not be displayed in Cross-Controller Dialog.

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

### See Also

`sceCrossControllerDialogParamInit()`, `SceCommonDialogParam`

# Obtainment of Operation Status



# sceCrossControllerDialogGetStatus

Get operation status of the Cross-Controller Dialog library

## Definition

```
#include <cross_controller_dialog.h>
SceCommonDialogStatus sceCrossControllerDialogGetStatus(void)
```

## 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	Cross-Controller Dialog is not running
SCE_COMMON_DIALOG_STATUS_RUNNING	0x1	Cross-Controller Dialog is running
SCE_COMMON_DIALOG_STATUS_FINISHED	0x2	Cross-Controller Dialog operation has finished

## Description

This function gets the operation status of the Cross-Controller Dialog library.

The default value of the operation status is SCE\_COMMON\_DIALOG\_STATUS\_NONE.

When calling sceCrossControllerDialogInit() is successful, the operation status immediately changes to SCE\_COMMON\_DIALOG\_STATUS\_RUNNING.

The operation status of the Cross-Controller Dialog library will not transition to SCE\_COMMON\_DIALOG\_STATUS\_FINISHED unless a PlayStation®3 is detected, the user cancels the operation, or until sceCrossControllerDialogAbort() is called.

sceCrossControllerDialogTerm() can be called only while the operation status is SCE\_COMMON\_DIALOG\_STATUS\_FINISHED.

When sceCrossControllerDialogTerm() is called, the operation status immediately changes to SCE\_COMMON\_DIALOG\_STATUS\_NONE.

This function is multithread safe.

## Examples

```
SceCommonDialogStatus stat = sceCrossControllerDialogGetStatus();
```

## See Also

```
sceCrossControllerDialogInit(), sceCrossControllerDialogAbort(),
sceCrossControllerDialogGetResult(), sceCrossControllerDialogTerm()
```

# **Obtainment of Cross-Controller Dialog Call Result**

SCE CONFIDENTIAL

# sceCrossControllerDialogGetResult

Get call result of Cross-Controller Dialog

## Definition

```
#include <cross_controller_dialog.h>
SceInt32
sceCrossControllerDialogGetResult(
    SceCrossControllerDialogResult *result
)
```

## Arguments

*result* 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 <i>result</i> argument
SCE_COMMON_DIALOG_ERROR_INVALID_ARGUMENT	0x80020403	Parameter error
SCE_COMMON_DIALOG_ERROR_NOT_FINISHED	0x80020410	Called during other than the appropriate operation status
SCE_COMMON_DIALOG_ERROR_NOT_IN_USE	0x80020411	sceCrossControllerDialogInit() is not called
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

## Description

This function obtains the call result of Cross-Controller Dialog.

This function can be called only while the operation status of Cross-Controller 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 sceCrossControllerDialogGetStatus() section.

The call result of Cross-Controller Dialog is stored in *result*. For details on the call result, refer to the SceCrossControllerDialogResult section.

This function is multithread safe.

SCE CONFIDENTIAL

---

**Examples**

---

```
SceCommonDialogStatus stat;

while(1) {
    stat = sceCrossControllerDialogGetStatus();
    if( stat == SCE_COMMON_DIALOG_STATUS_FINISHED ) {
        SceCrossControllerDialogResult result;
        sceClibMemset( &result, 0, sizeof(result) );
        if( 0 > sceCrossControllerDialogGetResult( &result ) ) {
            // error
        }
        sceCrossControllerDialogTerm();
        break;
    }
}
```

**See Also**

---

SceCrossControllerDialogResult, sceCrossControllerDialogGetStatus()

SCE CONFIDENTIAL

# SceCrossControllerDialogResult

Structure for obtaining Cross-Controller Dialog call result

## Definition

```
#include <cross_controller_dialog.h>
typedef struct SceCrossControllerDialogResult {
    SceInt32 result;
    SceNetInAddr inaddr;
    SceUInt8 reserved[24];
} SceCrossControllerDialogResult;
```

## Members

*result* Call result  
*inaddr* IP address of detected PlayStation®3  
*reserved* Reserved area

## Description

This structure receives the Cross-Controller Dialog call result. It is passed to `sceCrossControllerDialogGetResult()`. Fill it with all 0s during initialization.

The call result of Cross-Controller 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	PlayStation®3 detected
SCE_COMMON_DIALOG_RESULT_USER_CANCELED	0x1	User performed cancel operation
SCE_COMMON_DIALOG_RESULT_ABORTED	0x2	Abortion through sceCrossControllerDialogAbort()

*inaddr* stores the IP address of the detected PlayStation®3.

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

## See Also

`sceCrossControllerDialogInit()`, `sceCrossControllerDialogAbort()`,  
`sceCrossControllerDialogGetResult()`, `SceCrossControllerDialogParam`

# Aborting Cross-Controller Dialog

SCE CONFIDENTIAL

# sceCrossControllerDialogAbort

Abort call of Cross-Controller Dialog

## Definition

```
#include <cross_controller_dialog.h>
SceInt32 sceCrossControllerDialogAbort(void)
```

## 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	sceCrossControllerDialogInit() is not called
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

## Description

This function aborts calling of Cross-Controller Dialog. This function is used to abort the Cross-Controller Dialog display, for example when an urgent interrupt must be processed.

It can be called at any time between when sceCrossControllerDialogInit() is called and sceCrossControllerDialogTerm() is called. If it is called at times other than the above, SCE\_COMMON\_DIALOG\_ERROR\_NOT\_IN\_USE is returned.

When calling is successful, Cross-Controller 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 sceCrossControllerDialogGetStatus() section.

When Cross-Controller Dialog is closed with this function, calling sceCrossControllerDialogGetResult() returns the following.

```
SceCrossControllerDialogResult.result : SCE_COMMON_DIALOG_RESULT_ABORTED
```

This function is multithread safe.

SCE CONFIDENTIAL

---

**Examples**

---

```
SceCommonDialogStatus stat;

while(1) {
    stat = sceCrossControllerDialogGetStatus();
    if( stat == SCE_COMMON_DIALOG_STATUS_RUNNING ) {
        if( need_abort ) {
            sceCrossControllerDialogAbort();
            break;
        }
    }
    else if( stat == SCE_COMMON_DIALOG_STATUS_FINISHED ) {
        sceCrossControllerDialogTerm();
        break;
    }
}
```

**See Also**

---

```
sceCrossControllerDialogGetStatus(), sceCrossControllerDialogGetResult()
```



# Termination of Cross-Controller Dialog

SCE CONFIDENTIAL

# sceCrossControllerDialogTerm

Terminate call of Cross-Controller Dialog

## Definition

```
#include <cross_controller_dialog.h>
SceInt32 sceCrossControllerDialogTerm(void)
```

## 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
SCE_COMMON_DIALOG_ERROR_NOT_IN_USE	0x80020411	sceCrossControllerDialogInit() is not called
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

## Description

This function terminates calling of Cross-Controller Dialog. Calling must be terminated with this function after Cross-Controller Dialog has been called with sceCrossControllerDialogInit().

This function can be called only while the operation status of Cross-Controller 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 Cross-Controller Dialog 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 sceCrossControllerDialogGetStatus() section.

This function is multithread safe.

## Examples

```
SceCommonDialogStatus stat;

while(1) {
    stat = sceCrossControllerDialogGetStatus();
    if( stat == SCE_COMMON_DIALOG_STATUS_FINISHED ) {
        sceCrossControllerDialogTerm();
        break;
    }
}
```

## See Also

sceCrossControllerDialogInit(), sceCrossControllerDialogGetStatus()

©SCEI

## Constants

000004892117

SCE CONFIDENTIAL

# Return Codes

## List of return codes returned by Cross-Controller 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	sceCrossControllerDialogInit() is not called
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error
SCE_CROSS_CONTROLLER_DIALOG_ERROR_NEED_UPDATE_VITA_SYS	0x80107404	System update required
SCE_CROSS_CONTROLLER_DIALOG_ERROR_NEED_UPDATE_VITA_APP	0x80107405	Title update required
SCE_CROSS_CONTROLLER_DIALOG_ERROR_NEED_UPDATE_PS3_SYS	0x80107406	PlayStation®3 system update required
SCE_CROSS_CONTROLLER_DIALOG_ERROR_NEED_UPDATE_PS3_APP	0x80107407	PlayStation®3 title update required