

# **GameCustomDataDialog Library Reference**

© 2015 Sony Computer Entertainment Inc.  
All Rights Reserved.  
SCE Confidential

## Table of Contents

<b>Game Custom Data ID.....</b>	<b>3</b>
SceNpGameCustomDataId .....	4
<b>Parameter Settings.....</b>	<b>5</b>
SceGameCustomDataDialogAddressInfo.....	6
SceGameCustomDataDialogAddressParam .....	7
SceGameCustomDataDialogAddressType.....	8
SceGameCustomDataDialogDataParam.....	9
SceGameCustomDataDialogMode .....	12
SceGameCustomDataDialogOptions .....	13
SceGameCustomDataDialogParam .....	14
SceGameCustomDataDialogPlatformInfo .....	16
SceGameCustomDataMultiLanguageString .....	17
sceGameCustomDataDialogParamInit.....	18
<b>Call/Termination of the Game Custom Data Dialog.....</b>	<b>19</b>
sceGameCustomDataDialogInit.....	20
sceGameCustomDataDialogTerm .....	21
<b>Aborting the Game Custom Data Dialog.....</b>	<b>22</b>
sceGameCustomDataDialogAbort.....	23
<b>Obtaining the Game Custom Data Dialog Operation Status.....</b>	<b>25</b>
sceGameCustomDataDialogGetStatus.....	26
<b>Obtaining the Game Custom Data Dialog Call Result.....</b>	<b>27</b>
SceGameCustomDataDialogOnlineIdList.....	28
SceGameCustomDataDialogResult.....	29
sceGameCustomDataDialogGetResult .....	31
<b>Constants .....</b>	<b>32</b>
Return Codes .....	33

# Game Custom Data ID

000004892117

SCE CONFIDENTIAL

---

# SceNpGameCustomDataId

---

Game custom data ID

## Definition

---

```
#include <np_common.h>
typedef SceUInt64 SceNpGameCustomDataId;
```

## Description

---

This type represents a game custom data ID.

For the game custom data content, refer to the "Game Custom Data Web API s Reference" document.

# Parameter Settings

SCE CONFIDENTIAL

# SceGameCustomDataDialogAddressInfo

## Game custom data dialog addressee details

### Definition

```
#include <game_custom_data_dialog.h>
#define SCE_GAME_CUSTOM_DATA_DIALOG_ADDRESS_USER_LIST_MAX_NUMBER (16)
typedef union {
    struct {
        const SceNpOnlineId *onlineIds;
        SceUInt32 onlineIdsCount;
    } UserSelectDisableAddress;
    struct {
        SceUInt32 onlineIdsMaxCount;
    } UserSelectEnableAddress;
} SceGameCustomDataDialogAddressInfo;
```

### Members

<i>onlineIds</i>	Array of target users
<i>onlineIdsCount</i>	Number of target users (16 or less)
<i>onlineIdsMaxCount</i>	Maximum number of target users (16 or less)

### Description

This is a union for specifying detailed information of addressees when sending a game custom data with the game custom data dialog.

To specify `SCE_GAME_CUSTOM_DATA_DIALOG_ADDRESS_TYPE_USERDISABLE` as the editing mode for the game custom data addressee list and disable user edits, set the members *onlineIds* and *onlineIdsCount* in the *UserSelectDisableAddress* structure.

For *onlineIds*, specify the address for the target user array allocated in advance. Memory area allocated for *onlineIds* must be held until the game custom data dialog is terminated with `sceGameCustomDataDialogTerm()`.

For *onlineIdsCount*, specify the number of addressees specified in *onlineIds*.

To specify `SCE_GAME_CUSTOM_DATA_DIALOG_ADDRESS_TYPE_USERENABLE` as the editing mode for the game custom data addressee list and enable user edits, set the *onlineIdsMaxCount* member of the *UserSelectEnableAddress* structure.

For *onlineIdsMaxCount*, specify the maximum number of addressees that users can arbitrarily specify.

In either editing modes for the game custom data addressee list, the maximum number of users that can be specified as addressees is

`SCE_GAME_CUSTOM_DATA_DIALOG_ADDRESS_USER_LIST_MAX_NUMBER`.

### See Also

`sceGameCustomDataDialogParamInit()`, `sceGameCustomDataDialogInit()`,  
`SceGameCustomDataDialogAddressParam`

# SceGameCustomDataDialogAddressParam

Game custom data dialog addressee parameters

## Definition

```
#include <game_custom_data_dialog.h>
typedef struct SceGameCustomDataDialogAddressParam {
    SceGameCustomDataDialogAddressType addressType;
    SceGameCustomDataDialogAddressInfo addressInfo;
} SceGameCustomDataDialogAddressParam;
```

## Members

*addressType* Addressee edit mode  
*addressInfo* Addressee details

## Description

This structure is for specifying the addressee parameters when calling `sceGameCustomDataDialogInit()` in send mode.

For *addressType*, set the edit mode of the game custom data addressee list. Specify one of the following values.

Value	(Number)	Description
SCE_GAME_CUSTOM_DATA_DIALOG_ADDRESS_TYPE_USERDISABLE	1	Disable addressee list editing
SCE_GAME_CUSTOM_DATA_DIALOG_ADDRESS_TYPE_USERENABLE	2	Enable addressee list editing

If `SCE_GAME_CUSTOM_DATA_DIALOG_ADDRESS_TYPE_USERDISABLE` is specified for *addressType*, the user cannot edit the addressee list set by the application.

When `SCE_GAME_CUSTOM_DATA_DIALOG_ADDRESS_TYPE_USERENABLE` is specified for *addressType*, the user can edit the addressee list when game custom data dialog is displayed.

For *addressInfo*, specify the detailed information of the addressees. For details, refer to the "SceGameCustomDataDialogAddressInfo" section.

## See Also

SceGameCustomDataDialogDataParam, SceGameCustomDataDialogAddressType, SceGameCustomDataDialogAddressInfo

SCE CONFIDENTIAL

# SceGameCustomDataDialogAddressType

Game custom data addressee list editing mode

## Definition

```
#include <game_custom_data_dialog.h>
typedef SceInt32 SceGameCustomDataDialogAddressType;
```

## Description

These constants represent the edit mode of the game custom data addressee list.

Value	(Number)	Description
SCE_GAME_CUSTOM_DATA_DIALOG_ADDRESS_TYPE_INVALID	0	Invalid mode/initial value
SCE_GAME_CUSTOM_DATA_DIALOG_ADDRESS_TYPE_USERDISABLE	1	Addressee list editing disabled
SCE_GAME_CUSTOM_DATA_DIALOG_ADDRESS_TYPE_USERENABLE	2	Addressee list editing enabled

When SCE\_GAME\_CUSTOM\_DATA\_DIALOG\_ADDRESS\_TYPE\_USERDISABLE is specified for the editing mode, the user cannot edit the addressee list set by the application.

When SCE\_GAME\_CUSTOM\_DATA\_DIALOG\_ADDRESS\_TYPE\_USERENABLE is specified for the edit mode, the user can edit the addressee list when the game custom data dialog is displayed.

## See Also

SceGameCustomDataDialogAddressParam



SCE CONFIDENTIAL

# SceGameCustomDataDialogDataParam

Game custom data send/receive data parameters

## Definition

```
#include <game_custom_data_dialog.h>
#define SCE_GAME_CUSTOM_DATA_DIALOG_MAX_USER_MESSAGE_LENGTH (512)
#define SCE_GAME_CUSTOM_DATA_DIALOG_MAX_URL_LENGTH (1024)
#define SCE_GAME_CUSTOM_DATA_DIALOG_MAX_ATTACHMENT_DETAIL_LENGTH (512)
typedef union {
    struct {
        const SceChar8 *userMessage;
        const SceChar8 *dataName;
        const SceGameCustomDataMultiLanguageString *dataNameMultiLang;
        SceUInt32 dataNameMultiLangNum;
        const SceChar8 *dataDetail;
        const SceGameCustomDataMultiLanguageString *dataDetailMultiLang;
        SceUInt32 dataDetailMultiLangNum;
        void *thumbnail;
        SceUInt32 thumbnailSize;
        void *data;
        SceUInt32 dataSize;
        SceInt32 expireMinutes;
        SceGameCustomDataDialogPlatformInfo availablePlatform;
        SceGameCustomDataDialogAddressParam addressParam;
        SceGameCustomDataDialogOptions options;
    } SendInfo;
    struct {
        const SceChar8 *userMessage;
        const SceChar8 *dataName;
        const SceGameCustomDataMultiLanguageString *dataNameMultiLang;
        SceUInt32 dataNameMultiLangNum;
        const SceChar8 *dataDetail;
        const SceGameCustomDataMultiLanguageString *dataDetailMultiLang;
        SceUInt32 dataDetailMultiLangNum;
        void *thumbnail;
        SceUInt32 thumbnailSize;
        const SceChar8 *dataUrl;
        SceBool withItemId;
        SceInt32 expireMinutes;
        SceGameCustomDataDialogPlatformInfo availablePlatform;
        SceGameCustomDataDialogAddressParam addressParam;
        SceGameCustomDataDialogOptions options;
    } SendUrlInfo;
    struct {
        SceGameCustomDataDialogOptions options;
        SceUChar8 reserved[156];
    } RecvInfo;
} SceGameCustomDataDialogDataParam;
```

## Members

<i>userMessage</i>	Application-specified message string (UTF-8, NULL-terminated, 512 characters or less) or NULL
<i>dataName</i>	data name ((UTF-8, 64 characters or less)
<i>dataNameMultiLang</i>	Array of multilanguage data name strings, or NULL
<i>dataNameMultiLangNum</i>	Number of multilanguage data strings (maximum 10)
<i>dataDetail</i>	Data description (UTF-8, NULL-terminated, 512 characters or less) or NULL
<i>dataDetailMultiLang</i>	Array of multilanguage data descriptions, or NULL
<i>dataDetailMultiLangNum</i>	Number of multilanguage data descriptions (maximum 10)
<i>thumbnail</i>	Thumbnail JPEG image
<i>thumbnailSize</i>	Size of <i>thumbnail</i> (maximum 160 KiB)
<i>data</i>	Game custom data body (maximum 1 MiB)
<i>dataSize</i>	Size of <i>data</i>
<i>dataUrl</i>	URL string to attach (ASCII, NULL-terminated, maximum 1024 bytes)
<i>withItemId</i>	Flag to set whether or not add item ID to the URL
<i>expireMinutes</i>	Expiration time (minutes)
<i>availablePlatform</i>	Available platforms
<i>addressParam</i>	Addressee parameters
<i>options</i>	Options
<i>reserved</i>	Reserved area (fill with 0's)

## Description

This data parameter union is required for sending or receiving game custom data using the game custom data dialog. When the display mode of the game custom data dialog is send mode, set *SendInfo*; when the mode is send mode (with URL attachment), set *SendUrlInfo*. When the mode is receive mode, set *RecvInfo*.

The memory area allocated for setting each parameter must be kept until the game custom data dialog is closed with `sceGameCustomDataDialogTerm()`.

### Displaying the Game Custom Data Dialog in Send Mode

Set each member of *SendInfo*.

An arbitrary message string can be specified for *userMessage*. The string must be in UTF-8, NULL-terminated, and have a maximum size of

`SCE_GAME_CUSTOM_DATA_DIALOG_MAX_USER_MESSAGE_LENGTH` (characters). Specify NULL when not setting a string.

For *dataName*, specify the game custom data name.

For *dataNameMultiLang* and *dataNameMultiLangNum*, specify an array of multilanguage strings for the game custom data name and the number of elements respectively. For details, refer to the "SceGameCustomDataMultiLanguageString" section. Specify NULL and 0 respectively when not specifying multilanguage strings for game custom data name.

For *dataDetail*, specify a description of the game custom data. The string must be in UTF-8, NULL-terminated, and have a maximum size of

`SCE_GAME_CUSTOM_DATA_DIALOG_MAX_ATTACHMENT_DETAIL_LENGTH` (characters). Specify NULL when not setting a string.

For *dataDetailMultiLang* and *dataDetailMultiLangNum*, specify an array of multilanguage strings for the game custom data description and the number of elements respectively. For details, refer to the "SceGameCustomDataMultiLanguageString" section. Specify NULL and 0 respectively when not specifying multilanguage string for game custom data description.

For *thumbnail*, specify a pointer to the memory area storing the JPEG image. This image will be displayed on the system UI when viewing game custom data.

For *thumbnailSize*, specify the size of the image specified to *thumbnail*.

For *data* and *dataSize*, specify the binary data of game custom data arbitrarily defined by the application and the size respectively.

For *expireMinutes*, specify the expiration time in minutes. No expiration time will be set when 0 is specified.

For *availablePlatform*, specify usable platforms. For details, refer to the "SceGameCustomDataDialogPlatformInfo" section.

For *addressParam*, specify the addressee parameters indicating the addressees of the game custom data. For details, refer to the "SceGameCustomDataDialogAddressParam" section.

For *option*, set options of the game custom data dialog. For details, refer to the "SceGameCustomDataDialogOptions" section.

### Displaying the Game Custom Data Dialog in Send Mode (with URL Attachment)

Set each member of *SendUrlInfo*.

For *dataUrl*, specify the URL string to attach. The string must be ASCII and NULL-terminated; its maximum size is `SCE_GAME_CUSTOM_DATA_DIALOG_MAX_URL_LENGTH` (characters).

For *withItemId*, specify whether or not to attach the item ID of the game custom data to the URL. Members other than the above are the same as *SendInfo*.

### Displaying the Game Custom Data Dialog in Receive Mode

Set *RecvInfo*.

For *option*, set options of the game custom data dialog. For details, refer to the "SceGameCustomDataDialogOptions" section.

*reserved* is a reserved area. This area must be filled with 0's.

### See Also

SceGameCustomDataDialogParam, SceGameCustomDataDialogAddressParam,  
SceGameCustomDataDialogOptions

SCE CONFIDENTIAL

---

# SceGameCustomDataDialogMode

---

Game custom data dialog display mode

## Definition

```
#include <game_custom_data_dialog.h>
typedef SceInt32 SceGameCustomDataDialogMode;
```

## Description

These constants indicate the display mode for the game custom data dialog.

Value	(Number)	Description
SCE_GAME_CUSTOM_DATA_DIALOG_MODE_INVALID	0	Invalid mode/initial value
SCE_GAME_CUSTOM_DATA_DIALOG_MODE_SEND	1	Send mode
SCE_GAME_CUSTOM_DATA_DIALOG_MODE_RECV	2	Receive mode
SCE_GAME_CUSTOM_DATA_DIALOG_MODE_SEND_URL	3	Send mode (with URL attachment)

## See Also

SceGameCustomDataDialogParam

SCE CONFIDENTIAL

# SceGameCustomDataDialogOptions

## Game custom data dialog options

### Definition

```
#include <game_custom_data_dialog.h>
typedef SceInt32 SceGameCustomDataDialogOptions;
```

### Description

These options can be specified upon sending/receiving game custom data using the game custom data dialog.

Set the behavior upon calling the game custom data dialog by specifying the bit OR of the following values to the *options* member of the *SceGameCustomDataDialogDataParam* structure.

Value	(Number)	Description
-	0x0	No option specification
SCE_GAME_CUSTOM_DATA_DIALOG_OPTION_ASSUME_SEND	0x1	In send mode, send game custom data without the user having to press the send button of the game custom data dialog
SCE_GAME_CUSTOM_DATA_DIALOG_OPTION_SUSPEND_BLOCK	0x2	Prevent transitioning to the suspended state while sending or receiving

When SCE\_GAME\_CUSTOM\_DATA\_DIALOG\_OPTION\_ASSUME\_SEND is specified, game custom data can be sent upon calling the game custom data dialog in the send mode without the user having to press the send button of the game custom data dialog. This option specification is equivalent to SCE\_NP\_MESSAGE\_DIALOG\_OPTION\_ASSUME\_SEND provided by the NP Message Dialog library.

When SCE\_GAME\_CUSTOM\_DATA\_DIALOG\_OPTION\_SUSPEND\_BLOCK is specified, PlayStation®Vita will not transition to the suspended state while game custom data is being sent/received.

### See Also

*SceGameCustomDataDialogDataParam*

# SceGameCustomDataDialogParam

## Game custom data dialog parameters

### Definition

```
#include <game_custom_data_dialog.h>
typedef struct SceGameCustomDataDialogParam {
    SceUInt32 sdkVersion;
    SceCommonDialogParam commonParam;
    SceGameCustomDataDialogMode mode;
    void * callbackArg;
    const SceGameCustomDataDialogDataParam *dataParam;
    SceUChar8 reserved[64];
} SceGameCustomDataDialogParam;
```

### Members

*sdkVersion* SDK version  
*commonParam* Common dialog base parameters  
*mode* Game custom data dialog display mode  
*callbackArg* Application-defined data or NULL  
*dataParam* Data parameters required for sending or receiving game custom data  
*reserved* Reserved area (fill with 0's)

### Description

This structure is for specifying parameters for the game custom data dialog when the dialog is displayed with `sceGameCustomDataDialogInit()`. Before using this structure, it must be initialized using `sceGameCustomDataDialogParamInit()`.

For *mode*, specify the display mode of the game custom data dialog. Specify one of the following values.

Value	(Number)	Description
<code>SCE_GAME_CUSTOM_DATA_DIALOG_MODE_SEND</code>	1	Send mode
<code>SCE_GAME_CUSTOM_DATA_DIALOG_MODE_RECV</code>	2	Receive mode
<code>SCE_GAME_CUSTOM_DATA_DIALOG_MODE_SEND_URL</code>	3	Send mode (with URL attachment)

An arbitrary pointer can be specified for *callbackArg*. The pointer specified here will be stored as-is in the *callbackArg* member of the `SceGameCustomDataDialogResult` structure when the call result is obtained with `sceGameCustomDataDialogGetResult()`. Specify NULL when not requiring application-defined data.

For *dataParam*, specify the data parameters (game custom data, addressees, etc.) required for sending or receiving game custom data. For details, refer to the "SceGameCustomDataDialogDataParam" section.

For *commonParam*, specify common parameters for the Common dialog.

Although the display status of the info bar should be specified to *commonParam.infoBarParam*, always specify NULL for this value as the info bar is always displayed in the game custom data dialog and the application cannot control its display status.

Although the background color in ARGB format should be specified to *commonParam.bgColor*, when using the game custom data dialog in a game, only 0 (completely transparent) or 255 (completely opaque) can be specified to *commonParam.bgColor.a* (alpha component of the background color).

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

SCE CONFIDENTIAL

---

*reserved* is a reserved area. This area must be filled with 0's.

**See Also**

---

SceGameCustomDataDialogDataParam, sceGameCustomDataDialogParamInit(),  
sceGameCustomDataDialogInit()

000004892117

SCE CONFIDENTIAL

---

# SceGameCustomDataDialogPlatformInfo

---

Platform specification parameter of the game custom data dialog

## Definition

---

```
#include <game_custom_data_dialog.h>
#define SCE_GAME_CUSTOM_DATA_DIALOG_AVAILABLE_PLATFORM_PS4 "PS4"
#define SCE_GAME_CUSTOM_DATA_DIALOG_AVAILABLE_PLATFORM_PSVITA "PSVITA"
typedef struct SceGameCustomDataDialogPlatformInfo {
    SceUInt32 count;
    const SceChar8 **platformName;
} SceGameCustomDataDialogPlatformInfo;
```

## Members

---

<i>count</i>	Number of platform names
<i>platformName</i>	Array of platform names (ASCII)

## Description

---

This structure is for specifying usable platforms.

## See Also

---

SceGameCustomDataDialogDataParam



---

# SceGameCustomDataMultiLanguageString

---

Multilanguage string parameter of the game custom data dialog

## Definition

---

```
#include <game_custom_data_dialog.h>
typedef struct SceGameCustomDataMultiLanguageString {
    const SceChar8 *language;
    const SceChar8 *str;
} SceGameCustomDataMultiLanguageString;
```

## Members

---

<i>language</i>	Target languages
<i>str</i>	Strings (UTF-8)

## Description

---

This structure is for specifying multilanguage strings.

For language codes that can be specified in *language*, refer to the "Game Custom Data Web APIs Reference" document.

For *str*, specify strings in languages specified to *language*.

## See Also

---

SceGameCustomDataDialogDataParam

SCE CONFIDENTIAL

# sceGameCustomDataDialogParamInit

Macro for initializing call parameters

## Definition

```
#include <game_custom_data_dialog.h>
#define sceGameCustomDataDialogParamInit sceGameCustomDataDialogParamInitialize
static inline
void sceGameCustomDataDialogParamInitialize(SceGameCustomDataDialogParam *param)
{
    sceClibMemset( param, 0x0, sizeof(SceGameCustomDataDialogParam) );
    _sceCommonDialogSetMagicNumber( &param->commonParam );
    param->sdkVersion = SCE_PSP2_SDK_VERSION;
    param->mode = SCE_GAME_CUSTOM_DATA_DIALOG_MODE_INVALID;
}
```

## Arguments

*param* Call parameters to initialize

## Return Values

None

## Description

This macro function initializes parameters for calling the game custom data dialog.

Before making individual settings to call parameters, always use this macro to initialize the structure. The appropriate SDK version will be set and the reserved area will be filled with 0's.

## Examples

```
SceGameCustomDataDialogParam param;
sceGameCustomDataDialogParamInit (&param);
```

## See Also

SceGameCustomDataDialogParam

# **Call/Termination of the Game Custom Data Dialog**

SCE CONFIDENTIAL

# sceGameCustomDataDialogInit

Call the game custom data dialog

## Definition

```
#include <game_custom_data_dialog.h>
SceInt32 sceGameCustomDataDialogInit(
    const SceGameCustomDataDialogParam *param
)
```

## Arguments

*param* Call parameters

## Return Values

Returns SCE\_OK (0) as the value of the function for normal termination.

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

Value	(Number)	Description
SCE_COMMON_DIALOG_ERROR_BUSY	0x80020401	Another Common dialog is running
SCE_COMMON_DIALOG_ERROR_NULL	0x80020402	NULL was specified for <i>param</i>
SCE_COMMON_DIALOG_ERROR_INVALID_ARGUMENT	0x80020403	The content of <i>param</i> is invalid
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

## Description

This function calls the game custom data dialog.

For *\*param*, specify the game custom data dialog parameter structure. Have the structure initialized in advance with `sceGameCustomDataDialogParamInit()` and set with the required parameters (operation mode, for example).

This function can only be called while other Common dialogs (including the game custom data dialog itself) are not called. When called at any other time, SCE\_COMMON\_DIALOG\_ERROR\_BUSY returns.

When the call of this function succeeds, the operation status will immediately transition to SCE\_COMMON\_DIALOG\_STATUS\_RUNNING. For details on operation statuses, refer to the "sceGameCustomDataDialogGetStatus" section and the "Common Dialog Overview" document.

## Examples

```
SceGameCustomDataDialogParam param;

sceGameCustomDataDialogParamInit( &param );
param.mode = SCE_GAME_CUSTOM_DATA_DIALOG_MODE_RECV;

// Display the game custom data dialog
ret = sceGameCustomDataDialogInit(&param) ;
if (ret < 0) {
    // Error handling
}
```

## See Also

SceGameCustomDataDialogParam, sceGameCustomDataDialogParamInit()

©SCEI

SCE CONFIDENTIAL

# sceGameCustomDataDialogTerm

Terminate the game custom data dialog

## Definition

```
#include <game_custom_data_dialog.h>
SceInt32 sceGameCustomDataDialogTerm()
```

## Arguments

None

## Return Values

Returns SCE\_OK (0) as the value of the function for normal termination.

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

Value	(Number)	Description
SCE_COMMON_DIALOG_ERROR_NOT_FINISHED	0x80020410	Called at a time other than during the appropriate operation status
SCE_COMMON_DIALOG_ERROR_NOT_IN_USE	0x80020411	sceGameCustomDataDialogInit() has not been called

## Description

This function terminates the game custom data dialog. The game custom data dialog must always be terminated with this function after being called with `sceGameCustomDataDialogInit()`.

This function can only be called while the operation status of the game custom data dialog is SCE\_COMMON\_DIALOG\_STATUS\_FINISHED. If the game custom data dialog has not been called, SCE\_COMMON\_DIALOG\_ERROR\_NOT\_IN\_USE returns. If this function is called at times other than the above, SCE\_COMMON\_DIALOG\_ERROR\_NOT\_FINISHED returns.

When the call of this function succeeds, the operation status will immediately transition to SCE\_COMMON\_DIALOG\_STATUS\_NONE.

For details on operation statuses, refer to the "sceGameCustomDataDialogGetStatus" section.

## Examples

```
SceCommonDialogStatus stat;

while(1) {
    stat = sceGameCustomDataDialogGetStatus();
    if( stat == SCE_COMMON_DIALOG_STATUS_FINISHED ) {
        sceGameCustomDataDialogTerm();
        break;
    }
}
```

## See Also

`sceGameCustomDataDialogInit()`, `sceGameCustomDataDialogGetStatus()`

# Aborting the Game Custom Data Dialog

SCE CONFIDENTIAL

# sceGameCustomDataDialogAbort

Abort call of the game custom data dialog

## Definition

```
#include <game_custom_data_dialog.h>
SceInt32 sceGameCustomDataDialogAbort()
```

## Arguments

None

## Return Values

Returns SCE\_OK (0) as the value of the function for normal termination.

Returns the following error code (a negative value) for an error.

Value	(Number)	Description
SCE_COMMON_DIALOG_ERROR_NOT_IN_USE	0x80020411	sceGameCustomDataDialogInit() has not been called

## Description

This function aborts the game custom data dialog. Use this function to immediately abort display of the game custom data dialog when, for example, interrupt processing must suddenly be carried out.

This function can be called at any time after calling sceGameCustomDataDialogInit() and before calling sceGameCustomDataDialogTerm(). If this function is called at times other than the above, SCE\_COMMON\_DIALOG\_ERROR\_NOT\_IN\_USE returns.

When the call of this function succeeds, the process to close the displayed game custom data dialog will start and the operation status will transition to SCE\_COMMON\_DIALOG\_STATUS\_FINISHED after close processing completes.

For details on operation statuses, refer to the "sceGameCustomDataDialogGetStatus" section.

When this function is used to close the game custom data dialog, the call result obtained with sceGameCustomDataDialogGetResult() will be SCE\_COMMON\_DIALOG\_RESULT\_ABORTED.

## Examples

```
SceCommonDialogStatus stat;

while(1) {
    stat = sceGameCustomDataDialogGetStatus();
    if( stat == SCE_COMMON_DIALOG_STATUS_RUNNING ) {
        if( need_abort ) {
            sceGameCustomDataDialogAbort();
            break;
        }
    }
    else if( stat == SCE_COMMON_DIALOG_STATUS_FINISHED ) {
        sceGameCustomDataDialogTerm();
        break;
    }
}
```

SCE CONFIDENTIAL

---

**See Also**

---

sceGameCustomDataDialogGetResult ()

000004892117



# Obtaining the Game Custom Data Dialog Operation Status

SCE CONFIDENTIAL

# sceGameCustomDataDialogGetStatus

Get game custom data dialog operation status

## Definition

```
#include <game_custom_data_dialog.h>
SceCommonDialogStatus sceGameCustomDataDialogGetStatus ()
```

## Arguments

None

## Return Values

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

Value	(Number)	Description
SCE_COMMON_DIALOG_STATUS_NONE	0	Game custom data dialog is not running
SCE_COMMON_DIALOG_STATUS_RUNNING	1	Game custom data dialog is running
SCE_COMMON_DIALOG_STATUS_FINISHED	2	Game custom data dialog operation has finished

## Description

This function obtains the game custom data dialog operation status.

The operation status will be SCE\_COMMON\_DIALOG\_STATUS\_NONE before the call of the game custom data dialog.

The operation status will transition to SCE\_COMMON\_DIALOG\_STATUS\_RUNNING when the call of the game custom data dialog succeeds with sceGameCustomDataDialogInit().

The operation status will transition to SCE\_COMMON\_DIALOG\_STATUS\_FINISHED a little while after the close of the game custom data dialog by user operation or sceGameCustomDataDialogAbort(). sceGameCustomDataDialogGetResult() and sceGameCustomDataDialogTerm() can only be called while the operation status is SCE\_COMMON\_DIALOG\_STATUS\_FINISHED.

The operation status will immediately transition to SCE\_COMMON\_DIALOG\_STATUS\_NONE when sceGameCustomDataDialogTerm() is called.

## See Also

```
sceGameCustomDataDialogInit(), sceGameCustomDataDialogAbort(),
sceGameCustomDataDialogGetResult(), sceGameCustomDataDialogTerm()
```

# Obtaining the Game Custom Data Dialog Call Result

SCE CONFIDENTIAL

---

# SceGameCustomDataDialogOnlineIdList

---

Online ID list handled by the game custom data dialog

## Definition

---

```
#include <game_custom_data_dialog.h>
#define SCE_GAME_CUSTOM_DATA_DIALOG_ADDRESS_USER_LIST_MAX_NUMBER (16)
typedef struct SceGameCustomDataDialogOnlineIdList {
    SceUInt32 count;
    SceNpOnlineId
    onlineId[SCE_GAME_CUSTOM_DATA_DIALOG_ADDRESS_USER_LIST_MAX_NUMBER];
} SceGameCustomDataDialogOnlineIdList;
```

## Members

---

*count*      Number of online IDs  
*onlineId*   Online IDs

## Description

---

This structure represents the list of online IDs handled as a batch by the game custom data dialog.

## See Also

---

SceGameCustomDataDialogResult

# SceGameCustomDataDialogResult

## Game custom data dialog call result

### Definition

```
#include <game_custom_data_dialog.h>
typedef struct SceGameCustomDataDialogResult {
    void * callbackArg;
    SceInt32 errorCode;
    SceCommonDialogResult result;
    SceGameCustomDataDialogOnlineIdList *sentOnlineIds;
    SceUChar8 reserved[32];
} SceGameCustomDataDialogResult;
```

### Members

<i>callbackArg</i>	Application-defined argument specified upon calling the game custom data dialog
<i>errorCode</i>	Game custom data dialog close status
<i>result</i>	Game custom data dialog call result
<i>sentOnlineIds</i>	List of addressees to whom message was actually sent or NULL
<i>reserved</i>	Reserved area (fill with 0's)

### Description

This structure is for obtaining the game custom data dialog call result with `sceGameCustomDataDialogGetResult()`. This structure must be used after all values are filled with 0's and the structure is initialized in advance.

In *callbackArg*, the value for the *callbackArg* member of the argument *param* that was specified when `sceGameCustomDataDialogInit()` was called will be stored as-is. This member can be used by applications for arbitrary purposes.

In *errorCode*, the error that occurred upon termination of the game custom data dialog will be stored. SCE\_OK (0) will be stored for normal termination, and a non-0 value will be stored for a fatal error. The main error codes are shown below. Note, however, that the application must not malfunction even if other error codes are returned.

For error codes, also refer to the list of return codes in the "NpWebApi Library Reference" document, `include/net/errno.h`, `include/libnetctl.h`, and the list of error codes in the "Game Custom Data Web APIs Reference" document.

Value	(Number)	Description
SCE_NP_MANAGER_ERROR_NEED_CALL_NETCHECK_DIALOG	0x8055050b	Started the Network Check dialog in the PSN <sup>SM</sup> online mode but the service state of the NP library is not set to the online state
SCE_NET_ERROR_EIPADDRCHANGED	0x804101a3	Connection to the network was disconnected
SCE_NET_ERROR_EINACTIVEDISABLED		
SCE_NET_CTL_ERROR_WIFI_DISABLED	0x80412113	Wi-Fi is off
SCE_NP_WEBAPI_SERVER_ERROR_GAME_CUSTOM_DATA_BLOCKED_USER_EXISTS	0x82205185	Attempted to send to a user registered to the blocked user list
SCE_GAME_CUSTOM_DATA_DIALOG_ERROR_LACK_OF_LIBHTTP_POOL_SIZE	0x80109301	Remaining amount of the libhttp memory pool is insufficient
SCE_GAME_CUSTOM_DATA_DIALOG_ERROR_LACK_OF_LIBSSL_POOL_SIZE	0x80109302	Remaining amount of the libssl memory pool is insufficient

*result* is the game custom data dialog call result. One of the following values will be stored.

Value	(Number)	Description
SCE_COMMON_DIALOG_RESULT_OK	0	User carried out one of the following "OK" operations - Send completed and the dialog closed - Performed operation to use game custom data and closed the dialog
SCE_COMMON_DIALOG_RESULT_USER_CANCELED	1	User carried out one of the following cancel operations - Closed the dialog with the close button at the upper right - Pressed the cancel button during a send in the SCE_GAME_CUSTOM_DATA_DIALOG_OPTION_ASSUME_SEND mode and closed the dialog
SCE_COMMON_DIALOG_RESULT_ABORTED	2	Aborted with <code>sceGameCustomDataDialogAbort()</code>

For *sentOnlineIds*, specify a pointer to the `SceGameCustomDataDialogOnlineIdList` structure. When the game custom data send processing by the game custom data dialog succeeds, the number of addressees to whom the send was actually performed and a list of their online IDs will be stored in *sentOnlineIds*. If the online IDs of the addressees to whom send processing was performed are not needed, specify NULL for *sentOnlineIds*.

*reserved* is a reserved area. This area must be filled with 0's.

#### See Also

`sceGameCustomDataDialogGetResult()`

SCE CONFIDENTIAL

# sceGameCustomDataDialogGetResult

Get game custom data dialog call result

## Definition

```
#include <game_custom_data_dialog.h>
SceInt32 sceGameCustomDataDialogGetResult(
    SceGameCustomDataDialogResult *result
)
```

## Arguments

*result* Destination to store the obtained call result

## Return Values

Returns SCE\_OK (0) as the value of the function for normal termination.

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

Value	(Number)	Description
SCE_COMMON_DIALOG_ERROR_NULL	0x80020402	NULL was specified to <i>result</i>
SCE_COMMON_DIALOG_ERROR_NOT_FINISHED	0x80020410	Called at a time other than during the appropriate operation status
SCE_COMMON_DIALOG_ERROR_NOT_IN_USE	0x80020411	sceGameCustomDataDialogInit() has not been called

## Description

This function obtains the game custom data dialog call result.

This function can only be called when the game custom data dialog operation status is SCE\_COMMON\_DIALOG\_STATUS\_FINISHED. Otherwise, SCE\_COMMON\_DIALOG\_ERROR\_NOT\_FINISHED will return.

For details on operation statuses, refer to the "sceGameCustomDataDialogGetStatus" section.

The game custom data dialog call result will be stored in *\*result* upon normal termination of this function. For details on the call result, refer to the "SceGameCustomDataDialogResult" section.

*\*result* must always be initialized before it is passed to this function.

## See Also

SceGameCustomDataDialogResult, sceGameCustomDataDialogGetStatus()

## Constants

000004892117



## Return Codes

List of return codes returned by the GameCustomDataDialog library

### Definition

Value	(Number)	Description
SCE_COMMON_DIALOG_ERROR_BUSY	0x80020401	Another Common dialog is in use
SCE_COMMON_DIALOG_ERROR_NULL	0x80020402	NULL was specified for argument
SCE_COMMON_DIALOG_ERROR_INVALID_ARGUMENT	0x80020403	The content of argument is invalid
SCE_COMMON_DIALOG_ERROR_NOT_FINISHED	0x80020410	Called at a time other than during the appropriate operation status
SCE_COMMON_DIALOG_ERROR_NOT_IN_USE	0x80020411	sceGameCustomDataDialogInit() has not been called
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error
SCE_NP_MANAGER_ERROR_NEED_CALL_NETCHECK_DIALOG	0x8055050b	Called without carrying out sign-in processing with the Network Check dialog

In addition, error codes of the NpWebApi library will return to *errorCode* of *SceGameCustomDataDialogResult* that can be obtained with *sceGameCustomDataDialogGetResult()*.

For error codes, also refer to the list of return codes of the "NpWebApi Library Reference" document, *include/net/errno.h*, *include/libnetctl.h*, and the list of error codes of the "Game Custom Data Web APIs Reference" document.