

# **near Dialog Utility Reference**

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

# Table of Contents

<b>Common Datatypes .....</b>	<b>4</b>
SceNearGiftInfo .....	5
SceNearGiftInfo2 .....	6
<b>Initialization and Termination Functions .....</b>	<b>7</b>
SceNearInitParam .....	8
sceNearUtilityInitialize .....	9
sceNearUtilityFinalize .....	11
<b>Setting Gift to be Distributed .....</b>	<b>12</b>
SceNearGiftId .....	13
SceNearGiftCondition .....	14
SceNearPlayerAttrs .....	15
SceNearGiftStatus .....	16
sceNearUtilitySetGift .....	17
sceNearUtilitySetGift2 .....	19
sceNearUtilityGetGift .....	21
sceNearUtilityGetGiftStatus .....	23
sceNearUtilityDeleteGift .....	24
<b>Obtainment/Deletion of Discovered Gifts .....</b>	<b>25</b>
SceNearGiftDiscoveringId .....	26
SceNearDiscoveredGiftStatus .....	27
sceNearUtilityGetDiscoveredGifts .....	28
sceNearUtilityDeleteDiscoveredGift .....	29
sceNearUtilityIgnoreDiscoveredGift .....	30
sceNearUtilityGetDiscoveredGiftSender .....	31
sceNearUtilityGetDiscoveredGiftInfo .....	32
sceNearUtilityGetDiscoveredGiftStatus .....	33
sceNearUtilityOpenDiscoveredGiftImage .....	34
sceNearUtilityReadDiscoveredGiftImage .....	35
sceNearUtilityCloseDiscoveredGiftImage .....	36
sceNearUtilityOpenReceivedGiftData .....	37
sceNearUtilityReadReceivedGiftData .....	38
sceNearUtilityCloseReceivedGiftData .....	40
<b>Start-up of "near" Application .....</b>	<b>41</b>
sceNearUtilityLaunchNearAppForUpdate .....	42
sceNearUtilityLaunchNearAppForDownload .....	43
<b>Obtainment of Nearby Users .....</b>	<b>44</b>
sceNearUtilityGetNeighbors .....	45
sceNearUtilityGetRecentNeighbors .....	46
sceNearUtilityGetNewNeighbors .....	47
sceNearUtilityGetLastNeighborFoundDateTime .....	48
<b>Conversion of "near" Gift Event Parameters .....</b>	<b>49</b>
sceNearUtilityConvertDiscoveredGiftParam .....	50
<b>Obtainment of Usage Status of "near" Application .....</b>	<b>52</b>

---

SceNearMyStatus .....	53
sceNearUtilityGetMyStatus .....	54
<b>Re-obtainment of Information of "near" Application .....</b>	<b>55</b>
sceNearUtilityRefresh .....	56
<b>"near" Dialog Feature .....</b>	<b>58</b>
SceNearDialogMode .....	59
SceNearDialogParam .....	60
SceNearDialogResult .....	61
sceNearDialogParamInit .....	64
sceNearDialogInit .....	65
sceNearDialogGetStatus .....	67
sceNearDialogAbort .....	68
sceNearDialogGetResult .....	69
sceNearDialogTerm .....	71
<b>Constants .....</b>	<b>72</b>
List of Constants .....	73
Return Codes .....	74

# Common Datatypes

000004892117

SCE CONFIDENTIAL

# SceNearGiftInfo

Character string information of gifts to be distributed

## Definition

```
#include <near_dialog_util.h>
typedef struct SceNearGiftInfo {
    SceUInt32 giftNameLength;
    char giftName[SCE_NEAR_GIFT_NAME_MAX_LENGTH];
    char padding1[1];
    SceUInt32 giftDescriptionLength;
    char giftDescription[SCE_NEAR_GIFT_DESCRIPTION_MAX_LENGTH];
    char padding2[2];
} SceNearGiftInfo;
```

## Members

<i>giftNameLength</i>	Byte count of the character string indicating the gift name stored in <i>giftName</i> .
<i>giftName</i>	Does not include Null character Character string indicating gift name. Displayed on the first line of each item on the "near" application's "Discoveries" screen.
<i>padding1</i>	Described in UTF-8 and Null character termination are required Unused
<i>giftDescriptionLength</i>	Byte count of the character string describing the gift stored in <i>giftDescription</i> . Does not include Null character
<i>giftDescription</i>	Character string describing the gift. Displayed on the second line of each item on the "near" application's "Discoveries" screen.
<i>padding2</i>	Described in UTF-8 and Null character termination are required Unused

## Description

This is character string information of the gifts to be distributed.

On the "Discoveries" screen of the "near" application, up to 45 characters of the character string set in *giftName* and 90 characters of the character string set in *giftDescription* are displayed.

Specify a character string of 1 byte or more in *giftName* and *giftDescription*.

## See Also

`sceNearUtilitySetGift()`, `sceNearUtilityGetDiscoveredGiftInfo()`

SCE CONFIDENTIAL

# SceNearGiftInfo2

Character string information of gifts to be distributed (supporting multiple languages)

## Definition

```
#include <near_dialog_util.h>
typedef struct SceNearGiftInfo2 {
    SceUInt32 lang;
    SceUInt32 giftNameLength;
    char giftName[SCE_NEAR_GIFT_NAME_MAX_LENGTH];
    char padding1[1];
    SceUInt32 giftDescriptionLength;
    char giftDescription[SCE_NEAR_GIFT_DESCRIPTION_MAX_LENGTH];
    char padding[2];
} SceNearGiftInfo2;
```

## Members

<i>lang</i>	A language code representing the language of the character string stored in <i>giftName</i> and <i>giftDescriptionLength</i>
<i>giftNameLength</i>	Byte count of the character string indicating the gift name stored in <i>giftName</i> .
<i>giftName</i>	Does not include Null character Character string indicating gift name. Displayed on the first line of each item on the "near" application's "Discoveries" screen.
<i>padding1</i>	Described in UTF-8 and Null character termination are required Unused
<i>giftDescriptionLength</i>	Byte count of the character string describing the gift stored in <i>giftDescription</i> . Does not include Null character
<i>giftDescription</i>	Character string describing the gift. Displayed on the second line of each item on the "near" application's "Discoveries" screen.
<i>padding</i>	Described in UTF-8 and Null character termination are required Unused

## Description

This is character string information of the gifts to be distributed.

On the "Discoveries" screen of the "near" application, up to 45 characters of the character string set in *giftName* and 90 characters of the character string set in *giftDescription* are displayed.

Specify a character string of 1 byte or more in *giftName* and *giftDescription*.

In *lang*, store the language code of the character strings stored in *giftName* and *giftDescription*. For the language codes, refer to the "Application Utility Reference" document.

## See Also

sceNearUtilitySetGift2()

# Initialization and Termination Functions

SCE CONFIDENTIAL

---

# SceNearInitParam

---

"near" utility feature initialization parameters

## Definition

---

```
#include <near_dialog_util.h>
typedef struct SceNearInitParam {
    ScePVoid workMemory;
    SceUInt32 workMemorySize;
} SceNearInitParam;
```

## Members

---

<i>workMemory</i>	Starting address of work memory
<i>workMemorySize</i>	Size of work memory

## Description

---

This is parameters for initializing the "near" utility feature.

Specify the work memory used by the library in *workMemory*.

In the work memory, specify a memory space with a byte count equal to or greater than SCE\_NEAR\_UTIL\_DEFAULT\_WORKMEMORY\_SIZE.

## See Also

---

`sceNearUtilityInitialize()`



# sceNearUtilityInitialize

Initialize the "near" utility feature

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityInitialize(
    const SceNpCommunicationId *pCommId,
    const SceNearInitParam *pParam,
    SceUInt32 version
);
```

## Arguments

*pCommId* [in]:  
SceNpCommunicationId for exchanging gifts

*pParam* [in]:  
Parameters necessary for initialization

*version* [in]:  
Version of the data exchanged via SceNpCommunicationId

## 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	(Number)	Description
SCE_NEAR_ERROR_INVALID_ARGUMENT	0x80104901	Invalid argument (NULL was specified)
SCE_NEAR_ERROR_INVALID_COMMUNICATION_ID	0x80104902	Invalid value of SceNpCommunicationId
SCE_NEAR_ERROR_ALREADY_INITIALIZED	0x80104903	Already initialized
SCE_NEAR_ERROR_NO_MEMORY	0x80104904	Work area allocation failed
SCE_NEAR_ERROR_NETWORK_TIME_NOT_INITIALIZED	0x80104921	Network time not initialized
SCE_NEAR_DIALOG_ERROR_IN_USE	0x80107e06	Dialog currently in use

## Description

This function initializes "near" utility feature.

Specifies the SceNpCommunicationId for the transfer of gifts among titles.

When this function is called, resources such as work memory will be allocated inside the library.

Also, an external process will be started up and all information relating to gifts/nearby user will be obtained on the work memory.

The numerical value specified in *version* is intended to verify the compatibility of data exchanged via the same SceNpCommunicationId. In the "near" utility feature, it is not possible to read data registered with a version larger than the numerical value specified in this argument.

Initialization cannot be performed twice.

After calling `sceNearUtilityFinalize()`, it is, however, possible to perform initialization again. At this time, it is possible to perform initialization with another SceNpCommunicationId specified.

SCE CONFIDENTIAL

---

If the network time (UTC) in PlayStation®Vita has not been initialized, SCE\_NEAR\_ERROR\_NETWORK\_TIME\_NOT\_INITIALIZED is returned. For handling methods, refer to the "Handling for When Errors Occur" section in the "near Dialog Utility Overview" document.

**Notes**

---

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

**See Also**

---

`sceNearUtilityFinalize()`

SCE CONFIDENTIAL

# sceNearUtilityFinalize

Terminate the "near" utility feature

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityFinalize(
    const SceNpCommunicationId *pCommId
);
```

## Arguments

*pCommId* [in]:  
SceNpCommunicationId for exchanging gifts

## 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	(Number)	Description
SCE_NEAR_ERROR_INVALID_ARGUMENT	0x80104901	Invalid argument (NULL was specified)
SCE_NEAR_ERROR_INVALID_COMMUNICATION_ID	0x80104902	SceNpCommunicationId value is invalid or is not the value specified at initialization
SCE_NEAR_ERROR_NOT_INITIALIZED	0x80104905	Not initialized
SCE_NEAR_DIALOG_ERROR_IN_USE	0x80107e06	Dialog currently in use

## Description

This function terminates the "near" utility feature.

Resources allocated when sceNearUtilityInitialize is called will be freed.

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

sceNearUtilityInitialize()

## Setting Gift to be Distributed

SCE CONFIDENTIAL

# SceNearGiftId

## ID of gift to be distributed

### Definition

```
#include <near_dialog_util.h>
typedef SceUInt32 SceNearGiftId;
```

### Description

This is the ID identifying gifts distributed on the "near" server.

This ID is unique for each `SceNpCommunicationId`.

The upper 8 bits of `SceNearGiftId` have the following meanings:

- |            |   |
|------------|---|
| 0x80000000 | HIGH: this gift can also be discovered by users who do not have the title using the gift's <code>SceNpCommunicationId</code><br>LOW: this gift cannot be discovered without the title using the gift's <code>SceNpCommunicationId</code>  |
| 0x40000000 | HIGH: the gift of this kind can be discovered any number of times, provided that the gift is distributed by a different user<br>LOW: if one gift of this kind has been discovered, no more discoveries of gifts of the same kind will be possible   |
| 0x20000000 | HIGH: gifts of this kind contain data input by the user. They can be sent and received when the user's account is not subjected to chat restrictions.<br>LOW: gifts of this kind do not contain data input by the user. They can be sent and received regardless of whether the user's account is subjected to chat restrictions. |
| 0x1F000000 | Reserved bits<br>Set these bits to LOW(0).  |

### See Also

`sceNearUtilitySetGift()`, `sceNearUtilitySetGift2()`

SCE CONFIDENTIAL

# SceNearGiftCondition

## Receipt conditions for gifts to be distributed

### Definition

```
#include <near_dialog_util.h>
typedef struct SceNearGiftCondition {
    SceUInt16 radius;
    SceUInt16 duration;
    char reserved1[16];
    SceDateTime toTime;
    SceNearPlayerAttrs receiverAttrs;
    SceUInt8 probability;
    char padding[3];
} SceNearGiftCondition;
```

### Members

<i>radius</i>	Distance [m] in which the gift can be discovered
<i>duration</i>	Validity period [hour] from the discovery of gift data during which the game program can use the gift data
<i>reserved1</i>	Reserved area (fill all with 0x00)
<i>toTime</i>	End time/date for gift distribution
<i>receiverAttrs</i>	Attributes of users who can discover the gift
<i>probability</i>	Probability [%] of gift discovery
<i>padding</i>	Unused

### Description

This is the structure that describes the conditions for gift discovery/receiving. The "near" application/"near" Dialog utility on the receiving side references this structure when discovering/receiving gifts.

Each condition will be determined with AND.

If 0 is specified as the value of *radius*, processing will assume that the maximum value has been specified on the "near" server side. The maximum value is decided on the server side. If a value exceeding the value set by the server is specified in *radius*, it will be limited on the server side.

If the value of *duration* is 0, the validity period will be determined on the "near" system side. If a value exceeding the validity period set by the system is specified in *duration*, it will be limited on the system side.

Set *toTime* with UTC. If all of the *toTime* values are 0, the distribution end time/date will be determined by the system. If a value exceeding the maximum period for holding the gift data set by the system is specified in *toTime*, it will be limited on the system side.

The maximum value of *probability* is 100 [%]. If this value is 0, processing will assume that the prescribed maximum value (100 [%]) has been specified.

For more information on conditions for receiving gifts, refer to the "near System Overview" document.

### See Also

`sceNearUtilitySetGift()`, `sceNearUtilitySetGift2()`, `SceNearPlayerAttrs`

SCE CONFIDENTIAL

# SceNearPlayerAttrs

Attributes of users who can discover gifts

## Definition

```
#include <near_dialog_util.h>
typedef struct SceNearPlayerAttrs {
    SceUInt32 playerRelation;
} SceNearPlayerAttrs;

#define SCE_NEAR_PLAYER_RELATION_FRIEND      (1)
#define SCE_NEAR_PLAYER_RELATION_PLAYER     (2)
```

## Members

*playerRelation* Condition flag of the relation with the user to whom the gift is to be distributed (friend/general user)

## Description

This structure is passed as the attributes of users who can discover a gift when a gift to be distributed is set with `sceNearUtilitySetGift()` or `sceNearUtilitySetGift2()`.

Specify this by adding the values of `SCE_NEAR_PLAYER_RELATION_FRIEND` and `SCE_NEAR_PLAYER_RELATION_PLAYER` with an OR.

If 0 is specified for this value (if, that is, no flag is raised), the system side will assume that all flags have been raised.

## See Also

`sceNearUtilitySetGift()`, `sceNearUtilitySetGift2()`, `SceNearGiftCondition`

SCE CONFIDENTIAL

# SceNearGiftStatus

States of gift that have been set

## Definition

```
#include <near_dialog_util.h>
typedef SceUInt32 SceNearGiftStatus;

#define SCE_NEAR_GIFT_STATUS_NOT_REGISTERED (0)
#define SCE_NEAR_GIFT_STATUS_REGISTERED (1)
#define SCE_NEAR_GIFT_STATUS_POSTED (2)
```

## Enumeration Values

Value	(Number)	Description
SCE_NEAR_GIFT_STATUS_NOT_REGISTERED	0	Gift is not registered
SCE_NEAR_GIFT_STATUS_REGISTERED	1	Gift is registered, but is not sent to the "near" server
SCE_NEAR_GIFT_STATUS_POSTED	2	Gift has been sent to the "near" server

## Description

These indicate the state of the gift to be distributed that have been set.

## See Also

sceNearUtilityGetGiftStatus()



SCE CONFIDENTIAL

# sceNearUtilitySetGift

Set gift to be distributed

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilitySetGift (
    SceNearGiftId giftId,
    const SceNearGiftInfo *pGiftInfo,
    SceUInt32 giftImageSize,
    const void *pGiftImage,
    SceUInt32 giftDataSize,
    const void *pGiftData,
    SceUInt32 units,
    const SceNearGiftCondition *pGiftCondition
);
```

## Arguments

<i>giftId</i>	[in]: Uniquely defined gift ID for each <i>SceNpCommunicationId</i>
<i>pGiftInfo</i>	[in]: Gift character string information
<i>giftImageSize</i>	[in]: Gift image data size
<i>pGiftImage</i>	[in]: Gift image data
<i>giftDataSize</i>	[in]: Gift data size
<i>pGiftData</i>	[in]: Gift data
<i>units</i>	[in]: Number of gifts to be distributed
<i>pGiftCondition</i>	[in]: Gift receipt conditions

## 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	(Number)	Description
SCE_NEAR_ERROR_INVALID_ARGUMENT	0x80104901	Invalid argument (NULL was specified) or 0 was specified to <i>giftImageSize</i> or <i>giftDataSize</i>
SCE_NEAR_ERROR_NOT_INITIALIZED	0x80104905	Not initialized
SCE_NEAR_ERROR_NO_MEMORY	0x80104904	Work area allocation failed
SCE_NEAR_ERROR_INVALID_GIFT_INFO	0x80104906	Value specified in <i>SceNearGiftInfo</i> is invalid (character string size is too large or 0 was specified)
SCE_NEAR_ERROR_IMAGE_SIZE_OVER	0x80104907	Value of <i>giftImageSize</i> is too large
SCE_NEAR_ERROR_DATA_SIZE_OVER	0x80104908	Value of <i>giftDataSize</i> is too large

## SCE CONFIDENTIAL

Value	(Number)	Description
SCE_NEAR_ERROR_GIFT_COUNT	0x80104909	Gift distribution count exceeds the specifiable value
SCE_NEAR_ERROR_INVALID_CONDITION	0x8010490a	Value specified in <i>pGiftCondition</i> is invalid (an out-of-range value has been set)
SCE_NEAR_ERROR_NETWORK_TIME_NOT_INITIALIZED	0x80104921	Network time not initialized
SCE_NEAR_ERROR_INVALID_IMAGE	0x80104923	Image data is out of specification
SCE_NEAR_ERROR_INVALID_GIFT_ID	0x80104925	Reserved bits have not been cleared
SCE_NEAR_DIALOG_ERROR_IN_USE	0x80107e06	Dialog currently in use

**Description**

This function sets the gifts to be distributed and gift receipt conditions.

Gifts that have already been set will be deleted and overwritten.

Gift image data are 128 x 128 256-color CLUT in PNG format, with a maximum size of 8 KiB (SCE\_NEAR\_GIFT\_IMAGE\_MAX\_SIZE).

An error will return if it is attempted to set an image data that is out of specification.

The maximum size of gift data is 100 KiB (SCE\_NEAR\_GIFT\_DATA\_MAX\_SIZE). However, the first 256 bytes (SCE\_NEAR\_GIFT\_DATA\_PARAM\_MAX\_SIZE) will constitute the area reserved for a part of "near" gift event parameters". For details, refer to the "near Dialog Utility Overview" document.

The maximum value of *units* is 2147483647 (SCE\_NEAR\_GIFT\_MAX\_COUNT). If you do not wish to limit the distribution gift count, specify 0xFFFFFFFF in *units*.

**Notes**

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

**See Also**

SceNearGiftId, SceNearGiftInfo, SceNearGiftCondition, sceNearUtilityGetGift(),  
sceNearUtilityDeleteGift()

SCE CONFIDENTIAL

## sceNearUtilitySetGift2

Set gift to be distributed (supporting character strings in multiple languages)

### Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilitySetGift2 (
    SceNearGiftId giftId,
    SceUInt32 giftInfoNum,
    const SceNearGiftInfo2 *pGiftInfoArray,
    SceUInt32 giftImageSize,
    const void *pGiftImage,
    SceUInt32 giftDataSize,
    const void *pGiftData,
    SceUInt32 units,
    const SceNearGiftCondition *pGiftCondition
);
```

### Arguments

<i>giftId</i>	[in]: Uniquely defined gift ID for each SceNpCommunicationId
<i>giftInfoNum</i>	[in]: Size of array of gift character string information
<i>pGiftInfoArray</i>	[in]: Array of gift character string information
<i>giftImageSize</i>	[in]: Gift image data size
<i>pGiftImage</i>	[in]: Gift image data
<i>giftDataSize</i>	[in]: Gift data size
<i>pGiftData</i>	[in]: Gift data
<i>units</i>	[in]: Number of gifts to be distributed
<i>pGiftCondition</i>	[in]: Gift receipt conditions

### 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	(Number)	Description
SCE_NEAR_ERROR_INVALID_ARGUMENT	0x80104901	Invalid argument NULL was specified or 0 was specified to <i>giftImageSize</i> , <i>giftDataSize</i> or <i>giftInfoNum</i> , or 0 or a value exceeding SCE_NEAR_GIFT_LANG_MAX_NUM was specified in <i>giftInfoNum</i>
SCE_NEAR_ERROR_NOT_INITIALIZED	0x80104905	Not initialized
SCE_NEAR_ERROR_NO_MEMORY	0x80104904	Work area allocation failed

SCE CONFIDENTIAL

Value	(Number)	Description
SCE_NEAR_ERROR_INVALID_GIFT_INFO2	0x80104926	Value specified in <code>SceNearGiftInfo2</code> is invalid (character string size is too large or 0 was specified)
SCE_NEAR_ERROR_IMAGE_SIZE_OVER	0x80104907	Value of <code>giftImageSize</code> is too large
SCE_NEAR_ERROR_DATA_SIZE_OVER	0x80104908	Value of <code>giftDataSize</code> is too large
SCE_NEAR_ERROR_GIFT_COUNT	0x80104909	Gift distribution count exceeds the specifiable value
SCE_NEAR_ERROR_INVALID_CONDITION	0x8010490a	Value specified in <code>pGiftCondition</code> is invalid (an out-of-range value has been set)
SCE_NEAR_ERROR_NETWORK_TIME_NOT_INITIALIZED	0x80104921	Network time not initialized
SCE_NEAR_ERROR_INVALID_IMAGE	0x80104923	Image data is out of specification
SCE_NEAR_ERROR_INVALID_GIFT_ID	0x80104925	Reserved bits have not been cleared
SCE_NEAR_DIALOG_ERROR_IN_USE	0x80107e06	Dialog currently in use
SCE_APPUTIL_ERROR_NOT_INITIALIZED	0x80100601	Application utility library has not been initialized

### Description

This function sets the gifts to be distributed and gift receipt conditions.

Gifts that have already been set will be deleted and overwritten.

Gift image data are 128 x 128 256-color CLUT in PNG format, with a maximum size of 8 KiB (SCE\_NEAR\_GIFT\_IMAGE\_MAX\_SIZE).

An error will return if it is attempted to set an image data that is out of specification.

The maximum size of gift data is 100 KiB (SCE\_NEAR\_GIFT\_DATA\_MAX\_SIZE). However, the first 256 bytes (SCE\_NEAR\_GIFT\_DATA\_PARAM\_MAX\_SIZE) will constitute the area reserved for a part of "near" gift event parameters". For details, refer to the "near Dialog Utility Overview" document.

Refer to the "near System Overview" document on which language is selected and transmitted to the user on the receiving side when setting character strings in multiple languages.

The maximum value of `units` is 2147483647 (SCE\_NEAR\_GIFT\_MAX\_COUNT). If you do not wish to limit the distribution gift count, specify 0xFFFFFFFF in `units`.

### Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

### See Also

`SceNearGiftId`, `SceNearGiftInfo2`, `SceNearGiftCondition`, `sceNearUtilityGetGift()`, `sceNearUtilityDeleteGift()`

SCE CONFIDENTIAL

# sceNearUtilityGetGift

Obtain information of gift that was last set

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityGetGift (
    SceNearGiftId *pGiftId,
    SceNearGiftInfo *pGiftInfo,
    SceUInt32 *pGiftImageSize,
    void *pGiftImage,
    SceUInt32 *pGiftDataSize,
    void *pGiftData,
    SceUInt32 *pUnits,
    SceNearGiftCondition *pGiftCondition
);
```

## Arguments

<i>pGiftId</i>	[out]: Set gift ID
<i>pGiftInfo</i>	[out]: Character string information of the set gift
<i>pGiftImageSize</i>	[inout]: The input value is the size of <i>pGiftImage</i> where the image data of the set gift is to be saved. The output value is the size of the data that is actually saved to <i>pGiftImage</i> . If 0 is specified as the input value, data acquisition is not performed and the size of the data is set to <i>pGiftImageSize</i> and returned.
<i>pGiftImage</i>	[out]: Image data of the set gift
<i>pGiftDataSize</i>	[inout]: The input value is the size of <i>pGiftData</i> where the data of the set gift is to be saved. The output value is the size of the data that is actually saved to <i>pGiftData</i> . If 0 is specified as the input value, data acquisition is not performed and the size of the data is set to <i>pGiftDataSize</i> and returned.
<i>pGiftData</i>	[out]: Data of the set gift
<i>pUnits</i>	[out]: If the gift distribution count has not been set to unlimited, this is the remaining distribution count, which is decremented on the "near" server
<i>pGiftCondition</i>	[out]: Conditions for receiving the set gift

SCE CONFIDENTIAL

## Return Values

Upon normal termination, returns 0 if there is no gift information that was last set, and returns 1 if gift information was obtained.

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

Value	(Number)	Description
SCE_NEAR_ERROR_INVALID_ARGUMENT	0x80104901	Invalid argument (NULL was specified)
SCE_NEAR_ERROR_NOT_INITIALIZED	0x80104905	Not initialized
SCE_NEAR_ERROR_IMAGE_RETRIEVE_FAILED	0x80104913	Saving of image data to buffer specified with <i>pGiftImageSize</i> failed
SCE_NEAR_ERROR_DATA_RETRIEVE_FAILED	0x80104914	Saving of data to buffer specified with <i>pGiftDataSize</i> failed
SCE_NEAR_DIALOG_ERROR_IN_USE	0x80107e06	Dialog currently in use

## Description

This function obtains the information of the gift to be distributed that was last set.

Specify NULL in the arguments for the information you do not wish to obtain.

Note that, however, *pGiftId* must be obtained without exception.

SCE\_NEAR\_ERROR\_INVALID\_ARGUMENT will be returned if NULL is specified.

When setting character strings in multiple languages to a gift by using *sceNearUtilitySetGift2()*, the character string in the language that is meant to be displayed on the receiving side based on system language settings is stored in *pGiftInfo*. Refer to the "near System Overview" document on which language is selected when setting character strings in multiple languages.

When a value other than 0 is specified as an input value of *pGiftImageSize*, NULL cannot be specified in *pGiftImage*.

When a value other than 0 is specified as an input value of *pGiftDataSize*, NULL cannot be specified in *pGiftData*.

*pUnits* is the value obtained when the "near" application last communicated with the "near" server. It is not the latest distribution count of gifts remaining on the server.

If the distribution count of gifts was set to unlimited, *pUnits* returns 0xFFFFFFFF.

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

*sceNearUtilitySetGift()*, *sceNearUtilitySetGift2()*, *SceNearGiftId*, *SceNearGiftInfo*, *SceNearGiftCondition*

SCE CONFIDENTIAL

# sceNearUtilityGetGiftStatus

Obtain gift state that was last set

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityGetGiftStatus (
    SceNearGiftId giftId,
    SceNearGiftStatus *pGiftStatus
);
```

## Arguments

<i>giftId</i>	[in]: Set gift ID
<i>pGiftStatus</i>	[out]: Set gift state

## 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	(Number)	Description
SCE_NEAR_ERROR_INVALID_ARGUMENT	0x80104901	Invalid argument (NULL was specified)
SCE_NEAR_ERROR_NOT_INITIALIZED	0x80104905	Not initialized
SCE_NEAR_DIALOG_ERROR_IN_USE	0x80107e06	Dialog currently in use

## Description

This function obtains the state of the gift to be distributed that was last set.

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

SceNearGiftId, SceNearGiftStatus

SCE CONFIDENTIAL

# sceNearUtilityDeleteGift

Delete gift that was last set

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityDeleteGift (
    SceNearGiftId giftId
);
```

## Arguments

*giftId* [in]:  
Gift ID that is uniquely defined for each SceNpCommunicationId

## 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	(Number)	Description
SCE_NEAR_ERROR_NOT_INITIALIZED	0x80104905	Not initialized
SCE_NEAR_ERROR_NO_MEMORY	0x80104904	Work area allocation failed
SCE_NEAR_ERROR_DELETION_FAILED	0x80104918	Gift specified with <i>giftId</i> does not exist
SCE_NEAR_DIALOG_ERROR_IN_USE	0x80107e06	Dialog currently in use

## Description

This function deletes the gift to be distributed that was last set.

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

sceNearUtilitySetGift(), sceNearUtilitySetGift2(), SceNearGiftId



# Obtainment/Deletion of Discovered Gifts

SCE CONFIDENTIAL

---

# SceNearGiftDiscoveringId

---

ID of discovered gift

## Definition

---

```
#include <near_dialog_util.h>
typedef SceUInt32 SceNearGiftDiscoveringId;
```

## Description

---

ID for identifying discovered gifts.

## See Also

---

```
sceNearUtilityGetDiscoveredGifts(), sceNearUtilityDeleteDiscoveredGift(),
sceNearUtilityIgnoreDiscoveredGift(), sceNearUtilityGetDiscoveredGiftSender(),
sceNearUtilityGetDiscoveredGiftInfo(),
sceNearUtilityGetDiscoveredGiftStatus(),
sceNearUtilityOpenDiscoveredGiftImage(),
sceNearUtilityReadDiscoveredGiftImage(),
sceNearUtilityCloseDiscoveredGiftImage(),
sceNearUtilityOpenReceivedGiftData(), sceNearUtilityReadReceivedGiftData(),
sceNearUtilityCloseReceivedGiftData(),
sceNearUtilityConvertDiscoveredGiftParam()
```

SCE CONFIDENTIAL

# SceNearDiscoveredGiftStatus

Datatype representing status of discovered gifts

## Definition

```
#include <near_dialog_util.h>
typedef SceUInt32 SceNearDiscoveredGiftStatus;

#define SCE_NEAR_GIFT_STATUS_DISCOVERED      (0)
#define SCE_NEAR_GIFT_STATUS_RECEIVED        (1)
#define SCE_NEAR_GIFT_STATUS_EXPIRED         (2)
```

## Enumeration Values

Value	(Number)	Description
SCE_NEAR_GIFT_STATUS_DISCOVERED	0	Gift has already been discovered but has not been received yet
SCE_NEAR_GIFT_STATUS_RECEIVED	1	Gift has already been received and is ready to use
SCE_NEAR_GIFT_STATUS_EXPIRED	2	Gift has already been discovered or received, but its validity period has expired

## Description

These represent the status of discovered gifts.

## See Also

sceNearUtilityGetDiscoveredGiftStatus()

SCE CONFIDENTIAL

# sceNearUtilityGetDiscoveredGifts

Obtain list of discovered gifts

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityGetDiscoveredGifts (
    SceUInt32 *pArraySize,
    SceNearGiftDiscoveringId **ppGiftDiscoveringIdArray
);
```

## Arguments

*pArraySize*

[inout]:

The input value is the number of elements that can be obtained with *ppGiftDiscoveringIdArray*.

The output value is the actual number of elements stored in *ppGiftDiscoveringIdArray*.

If 0 is specified as the input value, the number of discovered gifts will be returned as return value.

*ppGiftDiscoveringIdArray*

[out]:

Address of the array of discovered gift's  
SceNearGiftDiscoveringId

## Return Values

Returns the number of gifts that can be returned as 0 or a positive value for normal termination.

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

Value	(Number)	Description
SCE_NEAR_ERROR_INVALID_ARGUMENT	0x80104901	Invalid argument (NULL was specified)
SCE_NEAR_ERROR_NOT_INITIALIZED	0x80104905	Not initialized
SCE_NEAR_DIALOG_ERROR_IN_USE	0x80107e06	Dialog currently in use

## Description

This function obtains a list of discovered gifts.

Use this function also when simply verifying the number of discovered gifts.

The maximum number of discovered gifts that can be obtained is

SCE\_NEAR\_GIFT\_DISCOVERED\_GIFT\_MAX\_NUM.

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

SceNearGiftDiscoveringId

SCE CONFIDENTIAL

# sceNearUtilityDeleteDiscoveredGift

## Delete discovered gifts

### Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityDeleteDiscoveredGift(
    SceNearGiftDiscoveringId giftDiscoveringId
);
```

### Arguments

*giftDiscoveringId* [in]:  
ID of the discovered gift obtained with  
`sceNearUtilityGetDiscoveredGifts()`

### Return Values

Returns `SCE_OK(0)` for success.

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

Value	(Number)	Description
<code>SCE_NEAR_ERROR_NOT_INITIALIZED</code>	0x80104905	Not initialized
<code>SCE_NEAR_ERROR_NO_MEMORY</code>	0x80104904	Work area allocation failed
<code>SCE_NEAR_ERROR_NOT_RETRIEVED</code>	0x8010491e	No gift has been discovered, or all discovered gifts are deleted
<code>SCE_NEAR_ERROR_DELETION_FAILED</code>	0x80104918	Gift specified with <i>giftDiscoveringId</i> does not exist
<code>SCE_NEAR_DIALOG_ERROR_IN_USE</code>	0x80107e06	Dialog currently in use

### Description

This function deletes discovered gifts from the "near" memory area.

Even if the user has set the gift to "Saved (Locked) Status" on the "near" application's "Discoveries" screen, the function will not check this and will delete the gift anyway.

### Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

### See Also

`sceNearUtilityGetDiscoveredGifts()`, `SceNearGiftDiscoveringId`

SCE CONFIDENTIAL

# sceNearUtilityIgnoreDiscoveredGift

Set discovered gift to the "Ignored" state

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityIgnoreDiscoveredGift(
    SceNearGiftDiscoveringId giftDiscoveringId
);
```

## Arguments

*giftDiscoveringId* [in]:  
ID of the discovered gift obtained with  
`sceNearUtilityGetDiscoveredGifts()`

## Return Values

Returns `SCE_OK(0)` for success.

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

Value	(Number)	Description
<code>SCE_NEAR_ERROR_NOT_INITIALIZED</code>	0x80104905	Not initialized
<code>SCE_NEAR_ERROR_NO_MEMORY</code>	0x80104904	Work area allocation failed
<code>SCE_NEAR_ERROR_NOT_RETRIEVED</code>	0x8010491e	No gift has been discovered, or all discovered gifts are deleted
<code>SCE_NEAR_ERROR_DELETION_FAILED</code>	0x80104918	Gift specified with <i>giftDiscoveringId</i> does not exist
<code>SCE_NEAR_DIALOG_ERROR_IN_USE</code>	0x80107e06	Dialog currently in use

## Description

This function sets discovered gifts to the "Ignored" state.

Even if the user has set the gift to "Saved (Locked) Status" on the "near" application's "Discoveries" screen, the function will not check this and will set it to the "Ignored" state. The state of gifts that have been set to the "Ignored" state will be managed by the "near" application. For details, refer to the "near System Overview" document.

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

`sceNearUtilityGetDiscoveredGifts()`, `SceNearGiftDiscoveringId`

SCE CONFIDENTIAL

# sceNearUtilityGetDiscoveredGiftSender

Obtain sender of discovered gifts

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityGetDiscoveredGiftSender (
    SceNearGiftDiscoveringId giftDiscoveringId,
    SceNpId *pNpId
);
```

## Arguments

*giftDiscoveringId* [in]:  
ID of the discovered gift obtained with  
`sceNearUtilityGetDiscoveredGifts()`

*pNpId* [out]:  
SceNpId of the discovered gift's sender

## Return Values

Returns SCE\_OK (0) for success.

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

Value	(Number)	Description
SCE_NEAR_ERROR_INVALID_ARGUMENT	0x80104901	Invalid argument (NULL was specified)
SCE_NEAR_ERROR_NOT_INITIALIZED	0x80104905	Not initialized
SCE_NEAR_ERROR_NOT_RETRIEVED	0x8010491e	No gift has been discovered, or all discovered gifts are deleted
SCE_NEAR_ERROR_RETRIEVE_FAILED	0x80104912	Gift specified with <i>giftDiscoveringId</i> does not exist
SCE_NEAR_DIALOG_ERROR_IN_USE	0x80107e06	Dialog currently in use

## Description

This function obtains the SceNpId of the sender of discovered gifts.

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

`sceNearUtilityGetDiscoveredGifts()`, `SceNearGiftDiscoveringId`

SCE CONFIDENTIAL

# sceNearUtilityGetDiscoveredGiftInfo

Obtain character string information of gifts discovered

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityGetDiscoveredGiftInfo (
    SceNearGiftDiscoveringId giftDiscoveringId,
    SceNearGiftInfo *pGiftInfo
);
```

## Arguments

*giftDiscoveringId* [in]:  
ID of the discovered gift obtained with  
`sceNearUtilityGetDiscoveredGifts()`

*pGiftInfo* [out]:  
Structure storing character string information of the discovered gift

## Return Values

Returns `SCE_OK (0)` for success.

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

Value	(Number)	Description
<code>SCE_NEAR_ERROR_INVALID_ARGUMENT</code>	0x80104901	Invalid argument (NULL was specified)
<code>SCE_NEAR_ERROR_NOT_INITIALIZED</code>	0x80104905	Not initialized
<code>SCE_NEAR_ERROR_NOT_RETRIEVED</code>	0x8010491e	No gift has been discovered, or all discovered gifts are deleted
<code>SCE_NEAR_ERROR_RETRIEVE_FAILED</code>	0x80104912	Gift specified with <i>giftDiscoveringId</i> does not exist
<code>SCE_NEAR_DIALOG_ERROR_IN_USE</code>	0x80107e06	Dialog currently in use

## Description

Structure storing character string information of the discovered gift

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

`sceNearUtilityGetDiscoveredGifts()`, `SceNearGiftDiscoveringId`, `SceNearGiftInfo`



SCE CONFIDENTIAL

# sceNearUtilityGetDiscoveredGiftStatus

Obtain storage status of discovered gifts

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityGetDiscoveredGiftStatus (
    SceNearGiftDiscoveringId giftDiscoveringId,
    SceNearDiscoveredGiftStatus *pGiftStatus
);
```

## Arguments

*giftDiscoveringId* [in]:  
ID of the discovered gift obtained with  
`sceNearUtilityGetDiscoveredGifts()`

*pGiftStatus* [out]:  
Storage status of discovered gifts

## Return Values

Returns `SCE_OK (0)` for success.

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

Value	(Number)	Description
<code>SCE_NEAR_ERROR_INVALID_ARGUMENT</code>	0x80104901	Invalid argument (NULL was specified)
<code>SCE_NEAR_ERROR_NOT_INITIALIZED</code>	0x80104905	Not initialized
<code>SCE_NEAR_ERROR_NOT_RETRIEVED</code>	0x8010491e	No gift has been discovered, or all discovered gifts are deleted
<code>SCE_NEAR_ERROR_RETRIEVE_FAILED</code>	0x80104912	Gift specified with <i>giftDiscoveringId</i> does not exist
<code>SCE_NEAR_DIALOG_ERROR_IN_USE</code>	0x80107e06	Dialog currently in use

## Description

This function obtains storage status of discovered gifts.

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

`sceNearUtilityGetDiscoveredGifts()`, `SceNearGiftDiscoveringId`,  
`SceNearDiscoveredGiftStatus`

SCE CONFIDENTIAL

# sceNearUtilityOpenDiscoveredGiftImage

Open image files of discovered gifts

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityOpenDiscoveredGiftImage (
    SceNearGiftDiscoveringId giftDiscoveringId
);
```

## Arguments

*giftDiscoveringId* [in]:  
ID of the discovered gift obtained with  
`sceNearUtilityGetDiscoveredGifts()`

## Return Values

Returns `SCE_OK (0)` for success.

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

Value	(Number)	Description
<code>SCE_NEAR_ERROR_NOT_INITIALIZED</code>	0x80104905	Not initialized
<code>SCE_NEAR_ERROR_NOT_RETRIEVED</code>	0x8010491e	No gift has been discovered, or all discovered gifts are deleted
<code>SCE_NEAR_ERROR_RETRIEVE_FAILED</code>	0x80104912	Gift specified with <i>giftDiscoveringId</i> does not exist
<code>SCE_NEAR_ERROR_FILE_OPEN_NUM</code>	0x80104916	Too many open files simultaneously
<code>SCE_NEAR_DIALOG_ERROR_IN_USE</code>	0x80107e06	Dialog currently in use

## Description

This function opens image files of discovered gifts.

The maximum number of gift image files that can be opened simultaneously is 3 (the number of open gift data is counted separately).

`SCE_NEAR_ERROR_RETRIEVE_FAILED` will be returned when the specified gift is deleted after the initialization of the library, in addition to the case where the value of *giftDiscoveringId* is not appropriate. This error will be returned when a gift is deleted with `sceNearUtilityDeleteDiscoveredGift()`, a gift is set to the "Ignored" state with `sceNearUtilityIgnoreDiscoveredGift()`, or a gift is deleted by "near" application in the background.

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

`sceNearUtilityGetDiscoveredGifts()`, `sceNearUtilityReadDiscoveredGiftImage()`,  
`sceNearUtilityCloseDiscoveredGiftImage()`, `SceNearGiftDiscoveringId`,  
`sceNearUtilityDeleteDiscoveredGift()`, `sceNearUtilityIgnoreDiscoveredGift()`

SCE CONFIDENTIAL

# sceNearUtilityReadDiscoveredGiftImage

Read image files of discovered gifts

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityReadDiscoveredGiftImage (
    SceNearGiftDiscoveringId giftDiscoveringId,
    void *pBuf,
    SceSize length,
    SceOff offset
);
```

## Arguments

*giftDiscoveringId* [in]:  
ID of the discovered gift specified with `sceNearUtilityOpenDiscoveredGiftImage()`

*pBuf* [out]:  
Buffer address to be read

*length* [in]:  
Byte count to be read  
When 0 is specified for this value, the total byte count of the file will return

*offset* [in]:  
Read start offset

## Return Values

Returns the number of bytes actually read or the total byte count as positive values for normal termination.

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

Value	(Number)	Description
SCE_NEAR_ERROR_INVALID_ARGUMENT	0x80104901	Invalid argument (NULL was specified)
SCE_NEAR_ERROR_NOT_INITIALIZED	0x80104905	Not initialized
SCE_NEAR_ERROR_FILE_NOT_OPENED	0x8010491b	File not opened
SCE_NEAR_ERROR_FILE_BAD_OFFSET	0x8010491d	Specified offset value is invalid
SCE_NEAR_DIALOG_ERROR_IN_USE	0x80107e06	Dialog currently in use

## Description

This function reads data from files opened with `sceNearUtilityOpenDiscoveredGiftImage()`. Allocate sufficient size for the read buffer on the caller side.

If 0 is specified in *length*, the file's total byte count will be checked and returned as a return value.

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

`sceNearUtilityGetDiscoveredGifts()`, `sceNearUtilityOpenDiscoveredGiftImage()`, `sceNearUtilityCloseDiscoveredGiftImage()`, `SceNearGiftDiscoveringId`

©SCEI

SCE CONFIDENTIAL

# sceNearUtilityCloseDiscoveredGiftImage

Close image files of discovered gifts

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityCloseDiscoveredGiftImage (
    SceNearGiftDiscoveringId giftDiscoveringId
);
```

## Arguments

*giftDiscoveringId* [in]:  
ID of the discovered gift specified with  
`sceNearUtilityOpenDiscoveredGiftImage()`

## Return Values

Returns `SCE_OK(0)` for success.

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

Value	(Number)	Description
<code>SCE_NEAR_ERROR_NOT_INITIALIZED</code>	0x80104905	Not initialized
<code>SCE_NEAR_ERROR_FILE_NOT_OPENED</code>	0x8010491b	File not opened
<code>SCE_NEAR_DIALOG_ERROR_IN_USE</code>	0x80107e06	Dialog currently in use

## Description

This function closes image files of discovered gifts.

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

`sceNearUtilityGetDiscoveredGifts()`, `sceNearUtilityOpenDiscoveredGiftImage()`,  
`sceNearUtilityReadDiscoveredGiftImage()`, `SceNearGiftDiscoveringId`

SCE CONFIDENTIAL

# sceNearUtilityOpenReceivedGiftData

Open data files of gifts received

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityOpenReceivedGiftData (
    SceNearGiftDiscoveringId giftDiscoveringId
);
```

## Arguments

*giftDiscoveringId* [in]:  
ID of the discovered gift obtained with  
`sceNearUtilityGetDiscoveredGifts()`

## 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	(Number)	Description
<code>SCE_NEAR_ERROR_NOT_INITIALIZED</code>	0x80104905	Not initialized
<code>SCE_NEAR_ERROR_NOT_RETRIEVED</code>	0x8010491e	No gift has been discovered, or all discovered gifts are deleted
<code>SCE_NEAR_ERROR_RETRIEVE_FAILED</code>	0x80104912	Gift specified with <i>giftDiscoveringId</i> does not exist
<code>SCE_NEAR_ERROR_FILE_OPEN_NUM</code>	0x80104916	Too many open files simultaneously
<code>SCE_NEAR_ERROR_NOT_RECEIVED</code>	0x80104919	Gift data not received yet
<code>SCE_NEAR_ERROR_EXPIRED</code>	0x8010491a	Gift validity period expired
<code>SCE_NEAR_DIALOG_ERROR_IN_USE</code>	0x80107e06	Dialog currently in use

## Description

This function opens data files of received gifts.

The maximum number of data files that can be opened simultaneously is 3 (the number of open gift images is counted separately).

`SCE_NEAR_ERROR_RETRIEVE_FAILED` will be returned when the specified gift is deleted after the initialization of the library, in addition to the case where the value of *giftDiscoveringId* is not appropriate. This error will be returned when a gift is deleted with `sceNearUtilityDeleteDiscoveredGift()`, a gift is set to the "Ignored" state with `sceNearUtilityIgnoreDiscoveredGift()`, or a gift is deleted by "near" application in the background.

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

`sceNearUtilityGetDiscoveredGifts()`, `sceNearUtilityReadReceivedGiftData()`,  
`sceNearUtilityCloseReceivedGiftData()`, `SceNearGiftDiscoveringId`,  
`sceNearUtilityDeleteDiscoveredGift()`, `sceNearUtilityIgnoreDiscoveredGift()`

©SCEI

SCE CONFIDENTIAL

# sceNearUtilityReadReceivedGiftData

Read data files of gifts received

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityReadReceivedGiftData (
    SceNearGiftDiscoveringId giftDiscoveringId,
    void *pBuf,
    SceSize length,
    SceOff offset
);
```

## Arguments

*giftDiscoveringId* [in]:  
ID of the discovered gift specified with `sceNearUtilityOpenReceivedGiftData()`

*pBuf* [out]:  
Buffer address to be read

*length* [in]:  
Byte count to be read

*offset* [in]:  
When 0 is specified for this value, the total byte count of the file will return  
Read start offset

## Return Values

Returns the number of bytes actually read or the total byte count as positive values for normal termination.

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

Value	(Number)	Description
SCE_NEAR_ERROR_INVALID_ARGUMENT	0x80104901	Invalid argument (NULL was specified)
SCE_NEAR_ERROR_NOT_INITIALIZED	0x80104905	Not initialized
SCE_NEAR_ERROR_FILE_NOT_OPENED	0x8010491b	File not opened
SCE_NEAR_ERROR_FILE_BAD_OFFSET	0x8010491d	Specified offset value is invalid
SCE_NEAR_DIALOG_ERROR_IN_USE	0x80107e06	Dialog currently in use

## Description

This function reads data from files opened with `sceNearUtilityOpenReceivedGiftData()`.  
Allocate sufficient size for the read buffer on the caller side.

If 0 is specified in the *length* argument, the file's total byte count will be checked and returned as a return value.

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

SCE CONFIDENTIAL

---

**See Also**

---

`sceNearUtilityGetDiscoveredGifts()`, `sceNearUtilityOpenReceivedGiftData()`,  
`sceNearUtilityCloseReceivedGiftData()`, `SceNearGiftDiscoveringId`

000004892117

SCE CONFIDENTIAL

# sceNearUtilityCloseReceivedGiftData

Close data files of gifts received

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityCloseReceivedGiftData (
    SceNearGiftDiscoveringId giftDiscoveringId
);
```

## Arguments

*giftDiscoveringId* [in]:  
ID of the discovered gift specified with  
`sceNearUtilityOpenReceivedGiftData()`

## Return Values

Returns `SCE_OK (0)` for success.

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

Value	(Number)	Description
<code>SCE_NEAR_ERROR_NOT_INITIALIZED</code>	0x80104905	Not initialized
<code>SCE_NEAR_ERROR_FILE_NOT_OPENED</code>	0x8010491b	File not opened
<code>SCE_NEAR_DIALOG_ERROR_IN_USE</code>	0x80107e06	Dialog currently in use

## Description

This function closes data files of received gifts.

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

`sceNearUtilityGetDiscoveredGifts()`, `sceNearUtilityOpenReceivedGiftData()`,  
`sceNearUtilityReadReceivedGiftData()`, `SceNearGiftDiscoveringId`



# Start-up of "near" Application

SCE CONFIDENTIAL

# sceNearUtilityLaunchNearAppForUpdate

Prompt information update by launching the "near" application

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityLaunchNearAppForUpdate (
    void
);
```

## Arguments

None

## Return Values

Returns SCE\_OK (0) for success.

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

Value	(Number)	Description
SCE_NEAR_ERROR_NO_MEMORY	0x80104904	Work area allocation failed
SCE_NEAR_DIALOG_ERROR_IN_USE	0x80107e06	Dialog currently in use

## Description

This function prompts the user to update the information by launching the "near" application and communicating with the "near" server.

This function can be called prior to "near" utility feature initialization with `sceNearUtilityInitialize()`.

If the information has been updated with the "near" application following initialization of the "near" utility feature, the information held in the memory on the library side is outdated, so either obtain the latest information using `sceNearUtilityRefresh()`, or discard the existing information with `sceNearUtilityFinalize()`.

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

`sceNearUtilityInitialize()`, `sceNearUtilityRefresh()`, `sceNearUtilityFinalize()`

SCE CONFIDENTIAL

# sceNearUtilityLaunchNearAppForDownload

Prompt receipt of discovered gifts by launching the "near" application

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityLaunchNearAppForDownload (
    SceNearGiftDiscoveringId giftDiscoveringId
);
```

## Arguments

*giftDiscoveringId* [in]:  
ID of discovered gift

## Return Values

Returns SCE\_OK (0) for success.

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

Value	(Number)	Description
SCE_NEAR_ERROR_NOT_INITIALIZED	0x80104905	Not initialized
SCE_NEAR_ERROR_NO_MEMORY	0x80104904	Work area allocation failed
SCE_NEAR_ERROR_NOT_RETRIEVED	0x8010491e	No gift has been discovered, or all discovered gifts are deleted
SCE_NEAR_ERROR_RETRIEVE_FAILED	0x80104912	Gift specified with <i>giftDiscoveringId</i> does not exist
SCE_NEAR_DIALOG_ERROR_IN_USE	0x80107e06	Dialog currently in use

## Description

This function prompts the user to receive discovered gifts by launching the "near" application.

Specify any of the discovered gifts with *giftDiscoveringId*.

If gift receipt has been performed with the "near" application following "near" utility feature initialization, the information held in the memory on the library side is outdated, so either obtain the latest information using `sceNearUtilityRefresh()`, or discard the existing information with `sceNearUtilityFinalize()`.

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

`sceNearUtilityInitialize()`, `sceNearUtilityRefresh()`, `sceNearUtilityFinalize()`

## Obtainment of Nearby Users

SCE CONFIDENTIAL

# sceNearUtilityGetNeighbors

Get list of nearby users

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityGetNeighbors (
    SceUInt32 *pArraySize,
    SceNpId **ppNpIdArray
);
```

## Arguments

*pArraySize* [inout]:  
The input value is the number of elements that can be received with *ppNpIdArray*  
The output value is the number of elements that was actually saved in *ppNpIdArray*  
If 0 is specified as the input value, the number of nearby users that can be obtained is returned as the return value.

*ppNpIdArray* [out]:  
Address of array of *SceNpId* of nearby user

## Return Values

Returns the number of nearby users that can be returned as 0 or a positive value upon normal completion.

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

Value	(Number)	Description
SCE_NEAR_ERROR_INVALID_ARGUMENT	0x80104901	Invalid argument (NULL was specified)
SCE_NEAR_ERROR_NOT_INITIALIZED	0x80104905	Not initialized
SCE_NEAR_DIALOG_ERROR_IN_USE	0x80107e06	Dialog currently in use

## Description

This function obtains the list of nearby users.

It is used also to simply find out the number of nearby users that can be obtained.

The maximum number of nearby users that can be obtained is

SCE\_NEAR\_GIFT\_NEIGHBORS\_MAX\_NUM.

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

sceNearUtilityInitialize()

SCE CONFIDENTIAL

# sceNearUtilityGetRecentNeighbors

Obtain a list of nearby users discovered at or after the specified time

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityGetRecentNeighbors (
    const SceRtcTick *pTime,
    SceUInt32 *pArraySize,
    SceNpId **ppNpIdArray
);
```

## Arguments

*pTime* [in]:  
Time to be used for narrowing the result

*pArraySize* [inout]:  
The input value is the number of elements that can be obtained with *ppNpIdArray*.  
The output value is the actual number of elements stored in *ppNpIdArray*.  
If 0 is specified as the input value, the number of obtainable nearby users will be returned as return value.

*ppNpIdArray* [out]:  
Address of array of *SceNpId* of nearby user

## Return Values

Returns the number of nearby users that can be returned as 0 or a positive value upon normal completion.

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

Value	(Number)	Description
SCE_NEAR_ERROR_INVALID_ARGUMENT	0x80104901	Invalid argument (NULL was specified)
SCE_NEAR_ERROR_NOT_INITIALIZED	0x80104905	Not initialized
SCE_NEAR_DIALOG_ERROR_IN_USE	0x80107e06	Dialog currently in use

## Description

This function obtains a list of nearby users discovered at or after the specified time.

The list returned from this function includes nearby users who have been discovered twice or more.

By specifying the time obtained with `sceNearUtilityGetLastNeighborFoundDateTime()`, it is possible to obtain nearby users only discovered through the most recent update processing of the "near" application.

This function can also be used only to get the number of obtainable nearby users.

The maximum number of nearby users that can be obtained is

`SCE_NEAR_GIFT_NEIGHBORS_MAX_NUM`.

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

`sceNearUtilityInitialize()`, `sceNearUtilityGetLastNeighborFoundDateTime()`

©SCEI

SCE CONFIDENTIAL

# sceNearUtilityGetNewNeighbors

Obtain a list of nearby users newly discovered at or after the specified time

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityGetNewNeighbors (
    const SceRtcTick *pTime,
    SceUInt32 *pArraySize,
    SceNpId **ppNpIdArray
);
```

## Arguments

*pTime* [in]:  
Time to be used for narrowing the result

*pArraySize* [inout]:  
The input value is the number of elements that can be obtained with *ppNpIdArray*.  
The output value is the actual number of elements stored in *ppNpIdArray*.  
If 0 is specified as the input value, the number of obtainable nearby users will be returned as return value.

*ppNpIdArray* [out]:  
Address of array of *SceNpId* of nearby user

## Return Values

Returns the number of nearby users that can be returned as 0 or a positive value upon normal completion.

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

Value	(Number)	Description
SCE_NEAR_ERROR_INVALID_ARGUMENT	0x80104901	Invalid argument (NULL was specified)
SCE_NEAR_ERROR_NOT_INITIALIZED	0x80104905	Not initialized
SCE_NEAR_DIALOG_ERROR_IN_USE	0x80107e06	Dialog currently in use

## Description

This function obtains a list of nearby users newly discovered at or after the specified time.

The list returned from this function excludes nearby users who have been discovered twice or more.

By specifying the time obtained with `sceNearUtilityGetLastNeighborFoundDateTime()`, it is possible to obtain nearby users only discovered through the most recent update processing of the "near" application.

This function can also be used only to get the number of obtainable nearby users.

The maximum number of nearby users that can be obtained is

`SCE_NEAR_GIFT_NEIGHBORS_MAX_NUM`.

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

`sceNearUtilityInitialize()`, `sceNearUtilityGetLastNeighborFoundDateTime()`

©SCEI

SCE CONFIDENTIAL

# sceNearUtilityGetLastNeighborFoundDateTime

Obtain the time at which the "near" application has last discovered a nearby user

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityGetLastNeighborFoundDateTime (
    SceRtcTick *pLastFoundTime
);
```

## Arguments

*pLastFoundTime* [out]:  
Time at which the "near" application has last discovered a nearby user

## Return Values

Returns SCE\_OK (0) for success.

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

Value	(Number)	Description
SCE_NEAR_ERROR_INVALID_ARGUMENT	0x80104901	Invalid argument (NULL was specified)
SCE_NEAR_ERROR_NOT_INITIALIZED	0x80104905	Not initialized
SCE_NEAR_ERROR_TIME_RETRIEVE_FAILED	0x80104924	Time data obtainment failed
SCE_NEAR_DIALOG_ERROR_IN_USE	0x80107e06	Dialog currently in use

## Description

This function obtains the time at which the "near" application has last discovered a nearby user.

There is a possibility that no nearby user has discovered as a result of update processing of the "near" application. In that case, the time returned by this function will not be updated.

SCE\_NEAR\_ERROR\_TIME\_RETRIEVE\_FAILED will return if obtainment of the time fails such as when the "near" application is not used or update processing has never been performed.

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

sceNearUtilityInitialize(), sceNearUtilityGetNewNeighbors(),  
sceNearUtilityGetRecentNeighbors()



# Conversion of "near" Gift Event Parameters

SCE CONFIDENTIAL

# sceNearUtilityConvertDiscoveredGiftParam

Convert the "near" gift event parameters and extract the required values

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityConvertDiscoveredGiftParam (
    SceAppUtilNearGiftParam *pGiftParam,
    SceNearGiftDiscoveringId *pGiftDiscoveringId
);
```

## Arguments

*pGiftParam* [in]:  
SceAppUtilNearGiftParam structure obtained by parsing "near" gift event parameters

*pGiftDiscoveringId* [out]:  
ID of the discovered gift obtained with `sceNearUtilityGetDiscoveredGifts()`

## 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	(Number)	Description
SCE_NEAR_ERROR_INVALID_ARGUMENT	0x80104901	Invalid argument (NULL was specified)
SCE_NEAR_ERROR_NOT_INITIALIZED	0x80104905	Not initialized
SCE_NEAR_ERROR_NOT_RETRIEVED	0x8010491e	No gift has been discovered, or all discovered gifts are deleted
SCE_NEAR_ERROR_RETRIEVE_FAILED	0x80104912	Gift corresponding to <i>pGiftParam</i> does not exist in the list of gifts obtained with <code>sceNearUtilityGetDiscoveredGifts()</code>
SCE_NEAR_ERROR_INVALID_VERSION	0x80104920	The version specified with the "near" gift event parameter is larger than the version given at the time of "near" utility feature initialization
SCE_NEAR_DIALOG_ERROR_IN_USE	0x80107e06	Dialog currently in use

## Description

As the information of the discovered gifts specified in the "Discoveries" screen of the "near" application has been saved to the "near" gift event parameters, this function is used to check `SceNearGiftDiscoveringId` from there.

The list of `SceNearGiftDiscoveringId` must be obtained with `sceNearUtilityGetDiscoveredGifts()`.

For details on the "near" gift event parameters, refer to the "near Dialog Utility Overview" document.

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

©SCEI

SCE CONFIDENTIAL

---

**See Also**

---

`sceNearUtilityInitialize(), SceNearGiftDiscoveringId`

000004892117

# **Obtainment of Usage Status of "near" Application**

SCE CONFIDENTIAL

---

# SceNearMyStatus

---

Usage status of "near" application

## Definition

---

```
#include <near_dialog_util.h>
typedef struct SceNearMyStatus {
    SceInt64 discoveredItemNum;
    SceInt64 discoveredTitleNum;
    SceInt64 encounterNum;
    SceDouble64 travelDistance;
} SceNearMyStatus;
```

## Members

---

<i>discoveredItemNum</i>	Cumulative number of discovered items
<i>discoveredTitleNum</i>	Cumulative number of discovered titles
<i>encounterNum</i>	Cumulative number of encounters
<i>travelDistance</i>	Cumulative travel distance [m]

## Description

---

This datatype indicates the usage status of the "near" application of the owner.

## See Also

---

`sceNearUtilityGetMyStatus()`

SCE CONFIDENTIAL

# sceNearUtilityGetMyStatus

Obtain usage status of "near" application

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityGetMyStatus (
    SceNearMyStatus *myStatus
);
```

## Arguments

*myStatus* [out]:  
Usage status of "near" application

## 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	(Number)	Description
SCE_NEAR_ERROR_INVALID_ARGUMENT	0x80104901	Invalid argument (NULL was specified)
SCE_NEAR_ERROR_NOT_INITIALIZED	0x80104905	Not initialized
SCE_NEAR_DIALOG_ERROR_IN_USE	0x80107e06	Dialog currently in use

## Description

This function obtains the usage status of the "near" application of the owner.

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

sceNearUtilityInitialize(), SceNearMyStatus

# **Re-obtainment of Information of "near" Application**

SCE CONFIDENTIAL

# sceNearUtilityRefresh

Obtain the latest information of the "near" application and save it to the work memory of the library

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearUtilityRefresh (
    const SceNpCommunicationId *pCommId
);
```

## Arguments

*pCommId* [in]:  
SceNpCommunicationId for exchanging gifts

## 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	(Number)	Description
SCE_NEAR_ERROR_INVALID_ARGUMENT	0x80104901	Invalid argument (NULL was specified)
SCE_NEAR_ERROR_INVALID_COMMUNICATION_ID	0x80104902	SceNpCommunicationId value is invalid or is not the value specified at initialization
SCE_NEAR_ERROR_NOT_INITIALIZED	0x80104905	Not initialized
SCE_NEAR_ERROR_NO_MEMORY	0x80104904	Work area allocation failed
SCE_NEAR_ERROR_NETWORK_TIME_NOT_INITIALIZED	0x80104921	Network time not initialized
SCE_NEAR_DIALOG_ERROR_IN_USE	0x80107e06	Dialog currently in use

## Description

This function discards the information held in the work memory with `sceNearUtilityInitialize()` and re-obtains the latest information of the "near" application.

Because the information obtained with the following functions prior to calling `sceNearUtilityRefresh()` becomes invalid when `sceNearUtilityRefresh()` is called, re-obtain the information by using the respective functions.

```
sceNearUtilityGetGift()
sceNearUtilityGetGiftStatus()
sceNearUtilityGetDiscoveredGifts()
sceNearUtilityGetDiscoveredGiftSender()
sceNearUtilityGetDiscoveredGiftInfo()
sceNearUtilityGetDiscoveredGiftStatus()
sceNearUtilityGetNeighbors()
sceNearUtilityGetRecentNeighbors()
sceNearUtilityGetNewNeighbors()
```



**SCE CONFIDENTIAL**

---

```
sceNearUtilityGetLastNeighborFoundDateTime()  
sceNearUtilityGetMyStatus()
```

Files that are opened with the following functions are automatically closed.

```
sceNearUtilityOpenDiscoveredGiftImage()  
sceNearUtilityOpenReceivedGiftData()
```

**Notes**

---

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

**See Also**

---

```
sceNearUtilityInitialize(), sceNearUtilityGetGift(),  
sceNearUtilityGetGiftStatus(), sceNearUtilityGetDiscoveredGifts(),  
sceNearUtilityGetDiscoveredGiftSender(),  
sceNearUtilityGetDiscoveredGiftInfo(),  
sceNearUtilityGetDiscoveredGiftStatus(), sceNearUtilityGetNeighbors(),  
sceNearUtilityGetRecentNeighbors(), sceNearUtilityGetNewNeighbors(),  
sceNearUtilityGetLastNeighborFoundDateTime(), sceNearUtilityGetMyStatus(),  
sceNearUtilityOpenDiscoveredGiftImage(), sceNearUtilityOpenReceivedGiftData()
```

## **"near" Dialog Feature**

000004892117

SCE CONFIDENTIAL

# SceNearDialogMode

enum constant specifying mode for calling "near" Dialog feature

## Definition

```
#include <near_dialog_util.h>
typedef enum SceNearDialogMode {
    SCE_NEAR_DIALOG_MODE_INVALID      = 0,
    SCE_NEAR_DIALOG_MODE_UPDATE       = 1,
    SCE_NEAR_DIALOG_MODE_UPDATE_RETRY = 2,
    SCE_NEAR_DIALOG_MODE_DOWNLOAD     = 3
} SceNearDialogMode;
```

## Description

These are modes for calling the features provided in the dialog format of the "near" Dialog utility.

Value	Description
SCE_NEAR_DIALOG_MODE_INVALID	Invalid mode
SCE_NEAR_DIALOG_MODE_UPDATE	Mode for communicating with the "near" server
SCE_NEAR_DIALOG_MODE_UPDATE_RETRY	Mode for retrying communication with the "near" server The time until location information obtainment timeout failure is long.
SCE_NEAR_DIALOG_MODE_DOWNLOAD	Mode for receiving distributed gifts

When the dialog is called with `sceNearDialogInit()`, this is saved to the *mode* member of the `SceNearDialogParam` structure.

## See Also

`sceNearDialogInit()`, `SceNearDialogParam`

SCE CONFIDENTIAL

# SceNearDialogParam

Parameters for calling "near" Dialog feature

## Definition

```
#include <near_dialog_util.h>
typedef struct SceNearDialogParam {
    SceUInt32 sdkVersion;
    SceNearDialogMode mode;
    SceNearGiftDiscoveringId giftDiscoveringId;
    SceChar8 reserved[32];
} SceNearDialogParam;
```

## Members

<i>sdkVersion</i>	SDK version
<i>mode</i>	Calling mode
<i>giftDiscoveringId</i>	ID of discovered gift
<i>reserved</i>	Reserved area (fill with all 0s)

## Description

These parameters are for calling the features provided in the dialog format of the "near" Dialog utility. Make sure to initialize this with `sceNearDialogParamInit()`.

Setting of the value to *sdkVersion* is done inside `sceNearDialogParamInit()`.

The value set to *giftDiscoveringId* is a meaningful value when the *mode* is `SCE_NEAR_DIALOG_MODE_DOWNLOAD`. When the *mode* is `SCE_NEAR_DIALOG_MODE_UPDATE`, this value is ignored.

## See Also

`sceNearDialogParamInit()`, `sceNearDialogInit()`, `SceNearDialogMode`

SCE CONFIDENTIAL

# SceNearDialogResult

Structure for obtaining "near" Dialog feature call result

## Definition

```
#include <near_dialog_util.h>
typedef struct SceNearDialogResult {
    SceUInt32 result;
    SceNearDialogMode mode;
    SceChar8 reserved[32];
} SceNearDialogResult;
```

## Members

*result* "near" Dialog feature execution result  
*mode* Calling mode  
*reserved* Reserved area (fill with all 0s)

## Description

This structure is used to obtain the result of the processing provided in the dialog format of the "near" Dialog utility. Fill it with all 0s during initialization.

The result of the "near" Dialog feature processing is saved to *result*.

The main error codes are shown below. Note, however, that the application must not malfunction even if other error codes are returned.

Value	(Number)	Description
SCE_COMMON_DIALOG_RESULT_OK	0x0	User selected a button
SCE_COMMON_DIALOG_RESULT_USER_CANCELED	0x1	User performed cancel operation
SCE_COMMON_DIALOG_RESULT_ABORTED	0x2	Aborted with <code>sceNearDialogAbort()</code>
SCE_NEAR_ERROR_INVALID_ARGUMENT	0x80104901	Invalid argument (NULL was specified)
SCE_NEAR_ERROR_INVALID_COMMUNICATION_ID	0x80104902	<code>SceNpCommunicationId</code> value is invalid or is not the value specified at initialization
SCE_NEAR_ERROR_NOT_INITIALIZED	0x80104905	Not initialized
SCE_NEAR_ERROR_NO_MEMORY	0x80104904	Work area allocation failed
SCE_NEAR_ERROR_NETWORK_TIME_NOT_INITIALIZED	0x80104921	Network time not initialized
SCE_NEAR_DIALOG_ERROR_PARAM	0x80107e01	Invalid parameter
SCE_NEAR_DIALOG_ERROR_IN_USE	0x80107e06	"near" Dialog feature currently in use
SCE_NEAR_DIALOG_RESULT_SEND_OK	0x80107e3f	Transmission was successful but whether reception succeeded or failed is unknown

SCE CONFIDENTIAL

Value	(Number)	Description
SCE_NEAR_DIALOG_RESULT_NOT_SIGNIN	0x80107e10	"near" function cannot be used because the user is not signed in
SCE_NEAR_DIALOG_RESULT_ALREADY_UPDATING	0x80107e11	Updating
SCE_NEAR_DIALOG_RESULT_LAUNCH_NEAR_NEED_INITIALIZATION	0x80107e12	"near" application launch was selected (because of incomplete initialization)
SCE_NEAR_DIALOG_RESULT_LAUNCH_NEAR_NEED_GIFT_DOWNLOAD	0x80107e13	"near" application launch was selected (because download is in progress)
SCE_NEAR_DIALOG_RESULT_LAUNCH_NEAR_NEED_USER_AGREEMENT	0x80107e14	"near" application launch was selected (because EULA not agreed to)
SCE_NEAR_DIALOG_RESULT_LAUNCH_NEAR_NEED_UPDATE_PERMISSION	0x80107e15	"near" application launch was selected (because <b>Do not ask my permission again</b> is set to off)
SCE_NEAR_DIALOG_RESULT_LAUNCH_NEAR_NEED_SHARED_USERID	0x80107e16	"near" application launch was selected (because ID is not shared)
SCE_NEAR_DIALOG_RESULT_LAUNCH_NEAR_NEED_SHARED_CONTENT	0x80107e17	"near" application launch was selected (because of privacy content)
SCE_NEAR_DIALOG_RESULT_LOCATION_NOT_PERMITTED	0x80107e18	Location data cannot be used
SCE_NEAR_DIALOG_RESULT_LOCATION_CANCELED	0x80107e19	Location calculation canceled
SCE_NEAR_DIALOG_RESULT_WIFI_OFF	0x80107e1a	Wi-Fi setting off
SCE_NEAR_DIALOG_RESULT_SEND_CANCELED	0x80107e1b	Transmission canceled
SCE_NEAR_DIALOG_RESULT_SERVER_MAINTENANCE	0x80107e1c	"near" server error (undergoing maintenance)
SCE_NEAR_DIALOG_RESULT_SERVER_END	0x80107e1d	"near" server error (service end)
SCE_NEAR_DIALOG_RESULT_DOWNLOAD_CANCELED	0x80107e1e	Download canceled
SCE_NEAR_DIALOG_RESULT_GIFT_NOT_FOUND	0x80107e1f	Gift not found
SCE_NEAR_DIALOG_RESULT_GIFT_CORRUPTED	0x80107e20	Downloaded data is corrupted
SCE_NEAR_DIALOG_RESULT_GIFT_EXPIRED	0x80107e21	Gift expired
SCE_NEAR_DIALOG_RESULT_BGPROC_DISCONNECTED	0x80107e22	Communication with background process was disconnected
SCE_NEAR_DIALOG_RESULT_MEMORY_CARD_NOT_INSERTED	0x80107e23	Memory card not inserted
SCE_NEAR_DIALOG_RESULT_SEND_FAILED_UGC	0x80107e25	Failed to send a gift containing data input by the user due to user account restrictions

SCE CONFIDENTIAL

---

For the handling method when a value other than above is returned, refer to the "Handling for When Errors Occur" section of the "near Dialog Utility Overview" document.

The value of `SceNearDialogParam.mode` specified with `sceNearDialogInit()` is saved in *mode*. It is used to determine which mode's call result this is.

**See Also**

---

`sceNearDialogInit()`, `sceNearDialogGetResult()`, `SceNearDialogParam`

000004892117

SCE CONFIDENTIAL

# sceNearDialogParamInit

Macro for "near" Dialog feature call parameter initialization

## Definition

```
#include <near_dialog_util.h>
static inline
void sceNearDialogParamInit (SceNearDialogParam *param)
{
    sceClibMemset(param, 0x0, sizeof(SceNearDialogParam));
    _sceCommonDialogSetMagicNumber( &param->commonParam );
    param->sdkVersin = SCE_PSP2_SDK_VERSION;
}
```

## Arguments

*param* [in]:  
"near" Dialog feature call parameter

## Return Values

None

## Description

This is a macro function for initializing the "near" Dialog feature call parameters.

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

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

sceNearDialogInit(), SceNearDialogParam



SCE CONFIDENTIAL

# sceNearDialogInit

Use "near" Dialog feature

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearDialogInit (
    const SceNearDialogParam *param
);
```

## Arguments

*param* [in]:  
Parameters for calling the "near" Dialog feature

## Return Values

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

Returns a negative value for errors. The main error codes are shown below. Note, however, that the application must not malfunction even if other error codes are returned.

Value	(Number)	Description
SCE_COMMON_DIALOG_ERROR_BUSY	0x80020401	Calling another Common Dialog feature
SCE_COMMON_DIALOG_ERROR_NULL	0x80020402	NULL was specified as the function's argument
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error
SCE_NEAR_ERROR_ALREADY_INITIALIZED	0x80104903	Called in initialized state (only when <i>mode</i> is SCE_NEAR_DIALOG_MODE_UPDATE)
SCE_NEAR_ERROR_NOT_INITIALIZED	0x80104905	Not initialized (only when <i>mode</i> is SCE_NEAR_DIALOG_MODE_DOWNLOAD)
SCE_NEAR_ERROR_RETRIEVE_FAILED	0x80104912	Gifts not found
SCE_NEAR_ERROR_NOT_RETRIEVED	0x8010491e	There are no discovered gifts, or all discovered gifts have been deleted
SCE_NEAR_DIALOG_ERROR_PARAM	0x80107e01	Invalid parameter
SCE_NEAR_DIALOG_ERROR_MODULE	0x80107e02	The required module is not loaded
SCE_NEAR_DIALOG_ERROR_INVALID_MODE	0x80107e03	Invalid mode
SCE_NEAR_DIALOG_ERROR_INVALID_SDK_VERSION	0x80107e04	Invalid SDK
SCE_NEAR_DIALOG_ERROR_INVALID_GIFT_STATUS	0x80107e05	Invalid Gift status

## Description

---

This function is used to start the dialog format processing.

This function can be called only when other Common Dialog features are not called (including the "near" Dialog utility 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 `sceNearDialogGetStatus()` section.

For the value to be set to *param*, refer to `SceNearDialogParam`.

To call this function by specifying `SCE_NEAR_DIALOG_MODE_UPDATE` in `SceNearDialogParam.mode`, call this function not in the state initialized with `sceNearUtilityInitialize()`. If initialization is done with `sceNearUtilityInitialize()`, call this function in the state where it has been returned to the uninitialized state with `sceNearUtilityFinalize()`. When the calling mode is `SCE_NEAR_DIALOG_MODE_UPDATE`, `SCE_NEAR_ERROR_ALREADY_INITIALIZED` is returned if this function was called in the state initialized with `sceNearUtilityInitialize()`.

To call this function by specifying `SCE_NEAR_DIALOG_MODE_DOWNLOAD` in `SceNearDialogParam.mode`, this must be done in the state initialized with `sceNearUtilityInitialize()`. If this function is called in the state where it has not been initialized with `sceNearUtilityInitialize()`, `SCE_NEAR_ERROR_NOT_INITIALIZED` is returned.

For details, refer to the "'near' Dialog Utility States and Feature Call Availability" chapter of the "near Dialog Utility Overview" document.

This function does not update the information in the work memory. To use the information after this function has been used, you must update the information in the work memory by calling `sceNearUtilityInitialize()` or `sceNearUtilityRefresh()`.

For the handling method when an error code other than indicated above is returned, refer to the "Handling for When Errors Occur" section of the "near Dialog Utility Overview" document.

## Notes

---

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

---

`sceNearDialogGetStatus()`, `sceNearDialogAbort()`, `sceNearDialogGetResult()`, `sceNearDialogTerm()`

SCE CONFIDENTIAL

# sceNearDialogGetStatus

Get operation status of "near" Dialog feature

## Definition

```
#include <near_dialog_util.h>
SceCommonDialogStatus sceNearDialogGetStatus ();
```

## Arguments

None

## Return Values

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

Value	(Number)	Description
SCE_COMMON_DIALOG_STATUS_NONE	0x01	Before use of "near" Dialog feature
SCE_COMMON_DIALOG_STATUS_RUNNING	0x02	"near" Dialog feature is in use
SCE_COMMON_DIALOG_STATUS_FINISHED	0x03	"near" Dialog feature use completed

## Description

This function is used to obtain the operation status of the dialog format processing.

The initial operation status is SCE\_COMMON\_DIALOG\_STATUS\_NONE.

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

When the dialog is closed either through user operation or by calling sceNearDialogAbort(), the operation status changes to SCE\_COMMON\_DIALOG\_STATUS\_FINISHED after a while. sceNearDialogGetResult() and sceNearDialogTerm() can be called only while the operation status is SCE\_COMMON\_DIALOG\_STATUS\_FINISHED.

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

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

sceNearDialogInit()

SCE CONFIDENTIAL

# sceNearDialogAbort

Abort "near" Dialog feature processing

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearDialogAbort ();
```

## Arguments

None

## Return Values

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

Returns a negative value for errors. The main error codes are shown below. Note, however, that the application must not malfunction even if other error codes are returned.

Value	(Number)	Description
SCE_COMMON_DIALOG_ERROR_NOT_IN_USE	0x80020411	"near" Dialog feature is not in use
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

## Description

This function aborts the dialog format processing.

It can be called at any time between when `sceNearDialogInit()` is called and `sceNearDialogTerm()` 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, processing is started to terminate the "near" Dialog feature being executed, and the operation status will change to `SCE_COMMON_DIALOG_STATUS_FINISHED` after the completion of the termination processing.

For details on the operation statuses, refer to the `sceNearDialogGetStatus()` section.

For the handling method when an error code other than indicated above is returned, refer to the "Handling for When Errors Occur" section of the "near Dialog Utility Overview" document.

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

`sceNearDialogInit()`

SCE CONFIDENTIAL

# sceNearDialogGetResult

Get processing result of "near" Dialog feature

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearDialogGetResult (
    SceNearDialogResult *result
);
```

## Arguments

*result* [out]:  
Result of "near" Dialog processing

## Return Values

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

Returns a negative value for errors. The main error codes are shown below. Note, however, that the application must not malfunction even if other error codes are returned.

Value	(Number)	Description
SCE_COMMON_DIALOG_ERROR_NULL	0x80020402	Invalid NULL pointer was specified in the argument
SCE_COMMON_DIALOG_ERROR_NOT_FINISHED	0x80020410	Terminating the "near" Dialog feature while it was running was attempted
SCE_NEAR_DIALOG_ERROR_PARAM	0x80107e01	Invalid parameter

## Description

This function obtains the result of the dialog format processing.

This function can be called only while the operation status of the "near" Dialog feature 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 sceNearDialogGetStatus() section.

The processing result of the "near" Dialog feature is stored in *result*. For details on the processing results, refer to the SceNearDialogResult section.

Be sure to initialize the argument *result* before passing it to this function. If *SceNearDialogResult.reserved* is not filled with 0s, SCE\_NEAR\_DIALOG\_ERROR\_PARAM is returned as the value of the function.

If the value of *result* is SCE\_NEAR\_DIALOG\_RESULT\_BGPROC\_DISCONNECTED, whether the processing has been completed is unknown. If required, retry this function.

An example of retry is given below.

If the result of the processing done by calling sceNearDialogInit() by setting SCE\_NEAR\_DIALOG\_MODE\_UPDATE in *SceNearDialogParam.mode* (the value of *SceNearDialogResult.result* obtained with sceNearDialogGetResult()) is SCE\_NEAR\_DIALOG\_RESULT\_BGPROC\_DISCONNECTED, call sceNearDialogTerm() to end the dialog format processing. You can retry by again setting SCE\_NEAR\_DIALOG\_MODE\_UPDATE in *SceNearDialogParam.mode* and calling sceNearDialogInit().

SCE CONFIDENTIAL

---

If the result of the processing done by calling `sceNearDialogInit()` by setting `SCE_NEAR_DIALOG_MODE_DOWNLOAD` in `SceNearDialogParam.mode` (the value of `SceNearDialogResult.result` obtained with `sceNearDialogGetResult()` is `SCE_NEAR_DIALOG_RESULT_BGPROC_DISCONNECTED`, call `sceNearDialogTerm()` to end the dialog format processing. You can retry by again setting `SCE_NEAR_DIALOG_MODE_DOWNLOAD` in `SceNearDialogParam.mode` and calling `sceNearDialogInit()`.

To check whether or not the processing was successful, after ending the dialog format processing by calling `sceNearDialogTerm()`, call `sceNearUtilityRefresh()` to update the information in the work memory, and obtain the state of the gifts to be downloaded by using `sceNearUtilityGetDiscoveredGifts()` and `sceNearUtilityGetDiscoveredGiftStatus()`. When `sceNearUtilityRefresh()` is called, the value of `SceNearGiftDiscoveringId` that was obtained prior to calling `sceNearUtilityRefresh()` becomes invalid, so caution is required when judging whether or not the processing was successful.

For the handling method when an error code other than above is stored in *result*, refer to the "Handling for When Errors Occur" section of the "near Dialog Utility Overview" document.

**Notes**

---

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

**See Also**

---

`sceNearDialogInit()`

SCE CONFIDENTIAL

# sceNearDialogTerm

Terminate "near" Dialog feature

## Definition

```
#include <near_dialog_util.h>
SceInt32 sceNearDialogTerm ();
```

## Arguments

None

## Return Values

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

Returns a negative value for errors. The main error codes are shown below. Note, however, that the application must not malfunction even if other error codes are returned.

Value	(Number)	Description
SCE_COMMON_DIALOG_ERROR_NOT_FINISHED	0x80020410	Terminating the "near" Dialog feature while it was running was attempted
SCE_COMMON_DIALOG_ERROR_NOT_IN_USE	0x80020411	The "near" Dialog feature is not in use

## Description

This function terminates use of the dialog format processing. Calling must be terminated with this function after the "near" Dialog feature has been called with `sceNearDialogInit()`.

This function can be called only while the operation status is SCE\_COMMON\_DIALOG\_STATUS\_FINISHED. If this function is called at 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 "near" Dialog feature 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 `sceNearDialogGetStatus()` section.

For the handling method when an error code other than indicated above is returned, refer to the "Handling for When Errors Occur" section of the "near Dialog Utility Overview" document.

## Notes

This function is not multithread safe. The operation is undefined when this function is called simultaneously from multiple threads.

## See Also

`sceNearDialogInit()`, `sceNearDialogGetStatus()`, `sceNearDialogAbort()`, `sceNearDialogGetResult()`

## Constants

000004892117



SCE CONFIDENTIAL

# List of Constants

## Macro constants

### Definition

Value	(Number)	Description
SCE_NEAR_UTIL_DEFAULT_WORKMEMORY_SIZE	262144	Size of the work memory used by the "near" utility feature
SCE_NEAR_GIFT_NAME_MAX_LENGTH	135	Maximum gift name size
SCE_NEAR_GIFT_DESCRIPTION_MAX_LENGTH	270	Maximum size of gift description
SCE_NEAR_GIFT_IMAGE_MAX_SIZE	8 * 1024	Maximum size of gift image data
SCE_NEAR_GIFT_DATA_MAX_SIZE	100 * 1024	Maximum size of gift data
SCE_NEAR_GIFT_DATA_PARAM_MAX_SIZE	256	Maximum size of data reserved as "near" gift event parameter
SCE_NEAR_GIFT_MAX_COUNT	2147483647	Maximum gift distribution count
SCE_NEAR_GIFT_DISCOVERED_GIFT_MAX_NUM	100	Maximum number of discovered gifts
SCE_NEAR_GIFT_NEIGHBORS_MAX_NUM	100	Maximum number of nearby users
SCE_NEAR_GIFT_LANG_MAX_NUM	20	Maximum number of languages that can be input as a gift's character string information

SCE CONFIDENTIAL

# Return Codes

List of return codes returned by the "near" Dialog utility

## Definition

Value	(Number)	Description
SCE_NEAR_ERROR_INVALID_ARGUMENT	0x80104901	Invalid argument (NULL was specified)
SCE_NEAR_ERROR_INVALID_COMMUNICATION_ID	0x80104902	Invalid value of <i>SceNpCommunicationId</i>
SCE_NEAR_ERROR_ALREADY_INITIALIZED	0x80104903	Already initialized
SCE_NEAR_ERROR_NO_MEMORY	0x80104904	Work area allocation failed
SCE_NEAR_ERROR_NOT_INITIALIZED	0x80104905	Not initialized
SCE_NEAR_ERROR_INVALID_GIFT_INFO	0x80104906	Value specified in <i>SceNearGiftInfo</i> is invalid (character string size is too large)
SCE_NEAR_ERROR_IMAGE_SIZE_OVER	0x80104907	Value of <i>giftImageSize</i> is too large
SCE_NEAR_ERROR_DATA_SIZE_OVER	0x80104908	Value of <i>giftDataSize</i> is too large
SCE_NEAR_ERROR_GIFT_COUNT	0x80104909	Gift distribution count exceeds the specifiable value
SCE_NEAR_ERROR_INVALID_CONDITION	0x8010490a	Value specified with <i>giftCondition</i> is invalid (an out-of-range value has been set)
SCE_NEAR_ERROR_RETRIEVE_FAILED	0x80104912	Gift information obtainment failed Information obtainment failed because specified gift does not exist
SCE_NEAR_ERROR_IMAGE_RETRIEVE_FAILED	0x80104913	Gift image data could not be obtained
SCE_NEAR_ERROR_DATA_RETRIEVE_FAILED	0x80104914	Gift data could not be obtained
SCE_NEAR_ERROR_FILE_OPEN_NUM	0x80104916	Too many open files simultaneously
SCE_NEAR_ERROR_DELETION_FAILED	0x80104918	Deletion failed because specified gift could not be found
SCE_NEAR_ERROR_NOT_RECEIVED	0x80104919	Gift data not received yet
SCE_NEAR_ERROR_EXPIRED	0x8010491a	Gift validity period expired
SCE_NEAR_ERROR_FILE_NOT_OPENED	0x8010491b	File not opened
SCE_NEAR_ERROR_FILE_BAD_OFFSET	0x8010491d	Specified offset value is invalid
SCE_NEAR_ERROR_NOT_RETRIEVED	0x8010491e	No gift has been discovered, or all discovered gifts are deleted
SCE_NEAR_ERROR_INVALID_VERSION	0x80104920	The version of the data to be read was larger than the version given at the time of utility initialization
SCE_NEAR_ERROR_NETWORK_TIME_NOT_INITIALIZED	0x80104921	Network time not initialized
SCE_NEAR_ERROR_INVALID_IMAGE	0x80104923	Image data is out of specification
SCE_NEAR_ERROR_TIME_RETRIEVE_FAILED	0x80104924	Time data obtainment failed
SCE_NEAR_DIALOG_ERROR_PARAM	0x80107e01	Invalid parameter
SCE_NEAR_DIALOG_ERROR_MODE	0x80107e02	The required module is not loaded
SCE_NEAR_DIALOG_ERROR_INVALID_MODE	0x80107e03	Invalid mode

SCE CONFIDENTIAL

Value	(Number)	Description
SCE_NEAR_DIALOG_ERROR_INVALID_SDK_VERSION	0x80107e04	Invalid SDK
SCE_NEAR_DIALOG_ERROR_INVALID_GIFT_STATUS	0x80107e05	Invalid Gift status
SCE_NEAR_DIALOG_ERROR_IN_USE	0x80107e06	"near" Dialog feature is in use
SCE_NEAR_DIALOG_RESULT_SEND_OK	0x80107e3f	Transmission was successful but whether reception succeeded or failed is unknown
SCE_NEAR_DIALOG_RESULT_NOT_SIGNIN	0x80107e10	"near" function cannot be used because the user is not signed in
SCE_NEAR_DIALOG_RESULT_ALREADY_UPDATING	0x80107e11	Updating
SCE_NEAR_DIALOG_RESULT_LAUNCH_NEAR_NEED_INITIALIZATION	0x80107e12	"near" application launch was selected (because of incomplete initialization)
SCE_NEAR_DIALOG_RESULT_LAUNCH_NEAR_NEED_GIFT_DOWNLOAD	0x80107e13	"near" application launch was selected (because download is in progress)
SCE_NEAR_DIALOG_RESULT_LAUNCH_NEAR_NEED_USER_AGREEMENT	0x80107e14	"near" application launch was selected (because EULA not agreed to)
SCE_NEAR_DIALOG_RESULT_LAUNCH_NEAR_NEED_UPDATE_PERMISSION	0x80107e15	"near" application launch was selected (because <b>Do not ask my permission again</b> is set to off)
SCE_NEAR_DIALOG_RESULT_LAUNCH_NEAR_NEED_SHARED_USERID	0x80107e16	"near" application launch was selected (because ID is not shared)
SCE_NEAR_DIALOG_RESULT_LAUNCH_NEAR_NEED_SHARED_CONTENT	0x80107e17	"near" application launch was selected (because of privacy content)
SCE_NEAR_DIALOG_RESULT_LOCATION_NOT_PERMITTED	0x80107e18	Location data cannot be used
SCE_NEAR_DIALOG_RESULT_LOCATION_CANCELED	0x80107e19	Location calculation canceled
SCE_NEAR_DIALOG_RESULT_WIFI_OFF	0x80107e1a	Wi-Fi setting off
SCE_NEAR_DIALOG_RESULT_SEND_CANCELED	0x80107e1b	Transmission canceled
SCE_NEAR_DIALOG_RESULT_SERVER_MAINTENANCE	0x80107e1c	"near" server error (undergoing maintenance)
SCE_NEAR_DIALOG_RESULT_SERVER_END	0x80107e1d	"near" server error (service end)
SCE_NEAR_DIALOG_RESULT_DOWNLOAD_CANCELED	0x80107e1e	Download canceled
SCE_NEAR_DIALOG_RESULT_GIFT_NOT_FOUND	0x80107e1f	Gifts not found
SCE_NEAR_DIALOG_RESULT_GIFT_CORRUPTED	0x80107e20	Downloaded data is corrupted
SCE_NEAR_DIALOG_RESULT_GIFT_EXPIRED	0x80107e21	Gift expired
SCE_NEAR_DIALOG_RESULT_BGPROC_DISCONNECTED	0x80107e22	Communication with background process was disconnected
SCE_NEAR_DIALOG_RESULT_MEMORY_CARD_NOT_INSERTED	0x80107e23	Memory card not inserted
SCE_NEAR_DIALOG_RESULT_UPDATE_RETRY	0x80107e24	Update retry was selected