NP Message Reference

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Table of Contents

Common Definitions

SceNpMessageId

Message ID

Definition

```
#include <np/np_message.h>
#define SCE_NP_MESSAGE_MAX_MESSAGE_ID_SIZE (70)
typedef struct SceNpMessageId {
        SceChar8 uid[SCE_NP_MESSAGE_MAX_MESSAGE_ID_SIZE];
        SceChar8 term;
        SceChar8 padding[1];
} SceNpMessageId;
```

Members

uid	Message ID
term	Terminator (specify NULL)
padding	Padding for adjusting alignment

Description

This structure represents a message ID.

It is used when obtaining a message body and data attached to a message.

`SCE_NP_MESSAGE_MAX_MESSAGE_ID_SIZE` is a constant representing the maximum size of a message ID

See Also

`SceNpMessageEntry`, `sceNpMessageGetMessage()`, `sceNpMessageGetAttachedData()`, `sceNpMessageSetAttachedDataUsedFlag()`

SceNpMessageType

Message type

Definition

```
#include <np/np_message.h>
typedef SceInt32 SceNpMessageType;
```

Description

This type represents a message type.

Message type represents the following types of messages:

Value	(Number)	Description
SCE_NP_MESSAGE_TYPE_INVALID	0x00	This constant represents an invalid message type.
SCE_NP_MESSAGE_TYPE_APP_DATA_ATTACHMENT	0x01	This constant represents a message with custom data attachments.
SCE_NP_MESSAGE_TYPE_INVITATION_ATTACHMENT	0x02	This constant represents a message with invitation data attachments.
SCE_NP_MESSAGE_TYPE_STORE_LINK_ATTACHMENT	0x03	This constant represents a message with store link attachments

See Also

sceNpMessageInit(),SceNpMessageParam

SceNpMessageEntry

Message data entry information

Definition

```
#include <np/np_message.h>
#define SCE_NP_MESSAGE_MAX_PREVIEW_SIZE (128)
typedef struct SceNpMessageEntry {
        SceNpMessageId messageId;
        SceDateTime dateTime;
        SceNpId from;
        SceChar8 preview[SCE_NP_MESSAGE_MAX_PREVIEW_SIZE];
        SceDateTime expiration_date;
        SceUInt16 messageFlag;
        SceChar8 reserved[30];
} SceNpMessageEntry;
```

Members

messageId	Message ID
dateTime	Date and time the message was sent (local time of PlayStation®Vita on receiving side)
from	NP ID of the message sender
preview	Beginning section of the message body
expiration_date	Message expiration date
messageFlag	Message flag
reserved	Reserved area

Description

This structure represents information about a message data entry.

It is used when obtaining a list of messages.

SCE_NP_MESSAGE_MAX_PREVIEW_SIZE is a constant representing the maximum size of the beginning section of the message body.

Message flag represents the following message statuses.

If a message is unread and no deletion processing has been performed, all bits will be 0 and the value of the message flag will be 0x00.

Value	(Number)	Description
SCE_NP_MESSAGE_FLAG_SEEN	0x02	This constant represents the message flag bits that have been read.
SCE_NP_MESSAGE_FLAG_DELETED	0x08	This constant represents the message flag bits that are to be deleted.
SCE_NP_MESSAGE_FLAG_USED	0x80	This constant represents message flag bits indicating used attached data.

See Also

SceNpMessageId, sceNpMessageGetMessageEntries()

SceNpMessageFilter

Message filter

Definition

```
#include <np/np_message.h>
typedef SceInt32 SceNpMessageFilter;
```

Description

This type represents a message filter.

By specifying the message filter, it is possible to display the messages corresponding to the specified filter conditions as a list, or to target them for obtaining processing. Specify the values in the following table by ORing.

Value	(Number)	Description
SCE_NP_MESSAGE_FILTER_NONE	0x0000	Among the messages that are sent to this application, only those that are unused and within the expiration date are included in this list. Normally this is the mode that is used.
SCE_NP_MESSAGE_FILTER_INCLUDE_USED	0x0001	In addition to SCE_NP_MESSAGE_FILTER_NONE, the messages that have been set as used with the sceNpMessageSetAttachedDataUsedFlag() function are included in the list.
SCE_NP_MESSAGE_FILTER_INCLUDE_EXPIRED	0x0002	In addition to SCE_NP_MESSAGE_FILTER_NONE, the messages whose expiration date has expired are also included in the list.

See Also

SceNpMessageParam

NP Message Dialog Call

sceNpMessageDialogParamInit

Macro for call parameter initialization

Definition

```
#include <np_message_dialog.h>
static inline
void sceNpMessageDialogParamInit(SceNpMessageDialogParam *param)
{
        sceClibMemset( param, 0x0, sizeof(SceNpMessageDialogParam) );
        _sceCommonDialogSetMagicNumber( &param->commonParam );
        param->sdkVersion = SCE_PSP2_SDK_VERSION;
        param->mode = SCE_NP_MESSAGE_DIALOG_MODE_INVALID;
}
```

Arguments

param	Call parameter to be initialized

Return Values

None

Description

This is a macro function for initializing the NP Message Dialog call parameter.

Before performing the various call parameter settings, be sure to use this macro to execute structure initialization. The appropriate SDK version is set.

Examples

```
// Initializing the parameter structure with sceNpMessageDialogParamInit()
SceNpMessageDialogParam      param;
sceNpMessageDialogParamInit(&param);
```

See Also

SceNpMessageDialogParam, SceNpMessageDialogMessageSendParam, SceNpMessageDialogMessageListParam, SceNpMessageDialogMessageDetailParam

sceNpMessageDialogInit

Call NP Message Dialog

Definition

```
#include <np_message_dialog.h>
SceInt32 sceNpMessageDialogInit(
        const SceNpMessageDialogParam *param,
)
```

Arguments

param   Call parameter

Return Values

Returns SCE_OK(0) as the value of the function for success.

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

Value	(Number)	Description
SCE_COMMON_DIALOG_ERROR_BUSY	0x80020401	The Common Dialog feature is already being called (details below)
SCE_COMMON_DIALOG_ERROR_NULL	0x80020402	NULL was specified for the *param* argument
SCE_COMMON_DIALOG_ERROR_INVALID_ARGUMENT	0x80020403	The content of the *param* argument is invalid
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

Description

This function calls NP Message Dialog.

This function can be called only when other Common Dialogs (including NP Message Dialog proper) are not called. 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 sceNpMessageDialogGetStatus() section.

In *param*, specify the call parameter structure for which the operation mode and operation type were set. Be sure to initialize the structure with sceNpMessageDialogParamInit() beforehand.

After being returned from this function, the *param* instance can be released. However, some parameters, including the instances specified by the *sendParam* member, must be held until NP Message Dialog feature is terminated by calling sceNpMessageDialogTerm().

This function is multithread safe.

Examples

```
// Example of use displaying NP Message Dialog in the message transmission mode
// Initialization of transmission parameter SceNpMessageDialogMessageSendParam
// structure
static SceNpMessageDialogMessageSendParam  s_sendParam;
sceClibMemset( &s_sendParam, 0x0,
                     sizeof(SceNpMessageDialogMessageSendParam) );

// Initialization of call parameter SceNpMessageDialogParam structure
SceNpMessageDialogParam     param;
sceNpMessageDialogParamInit(&param);

// Call parameter setting
param.mode            = SCE_NP_MESSAGE_DIALOG_MODE_MESSAGE_SEND;
param.type            = SCE_NP_MESSAGE_DIALOG_TYPE_APP_DATA_ATTACHMENT;
param.sendParam       = &s_sendParam;

// Setting of pointer to the array that includes SceNpId to be used as the
recipient
// and the number of SceNpId stored to the transmission parameter
static SceNpId       s_npIds[SCE_NP_MESSAGE_DIALOG_MAX_RECIPIENTS];
s_sendParam.npIds    = s_npIds;
s_sendParam.npIdsCount= count;      // Number of recipients stored in s_npIds[]
// If no recipient is specified, set NULL for .npIds and 0 for .npIdsCount

// Setting of message body to be transmitted
s_sendParam.body     = "Test Message";

// Setting of data to be attached to message to be transmitted
static const SceChar8 s_data[]     = "abcdedg";
s_sendParam.data     = (void*)s_data;
s_sendParam.dataSize = sizeof(s_data);

// Setting of index icon of message
s_sendParam.indexIconPath   = "app0:index_icon.jpg";

// Setting of expiration date of message
s_sendParam.expireMinutes   = 30;  // 30 minutes

// Display NP Message Dialog
ret = sceNpMessageDialogInit(&param) ;
if (ret < 0) {
        // Error handling
}
```

See Also

SceNpMessageDialogParam, sceNpMessageDialogParamInit(), sceNpMessageDialogGetStatus(), SceNpMessageDialogMessageSendParam, SceNpMessageDialogMessageListParam, SceNpMessageDialogMessageDetailParam

SceNpMessageDialogParam

NP Message Dialog call parameter

Definition

```
#include <np_message_dialog.h>
typedef struct SceNpMessageDialogParam {
        SceUInt32 sdkVersion;
        SceCommonDialogParam commonParam;
        SceNpMessageDialogMode mode;
        SceNpMessageDialogType type;
        SceNpMessageDialogOptions options;
        void *userData;
        SceNpMessageDialogMessageSendParam *sendParam;
        SceNpMessageDialogMessageListParam *listParam;
        SceNpMessageDialogMessageDetailParam *detailParam;
        SceNpMessageDialogMessageGetParam *getParam;
        SceChar8 reserved[28];
} SceNpMessageDialogParam;
```

Members

sdkVersion	SDK version
commonParam	Common parameters for Common Dialogs
mode	Operation mode (details below)
type	Message type
options	Dialog option
userData	User data
sendParam	Message transmission dialog parameter
listParam	Received message list dialog parameter
detailParam	Received message details dialog parameter
getParam	Message data obtainment dialog parameter
reserved	Reserved area (fill with all 0s)

Description

This is a structure passed to `sceNpMessageDialogInit()` to display NP Message Dialog. First, use `sceNpMessageDialogParamInit()` to initialize it and set a value to each member.

Specify the SDK version in *sdkVersion*. Since an appropriate value is input when the structure is initialized with `sceNpMessageDialogParamInit()`, do not change the value.

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

Specify the info bar information in *commonParam.infobarParam*. Always specify NULL since the info bar is always displayed in NP Message Dialog.

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

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

Specify the operation mode in *mode*. One of the following values must be specified.

Value	(Number)	Description
SCE_NP_MESSAGE_DIALOG_MODE_MESSAGE_SEND	1	Message transmission mode
SCE_NP_MESSAGE_DIALOG_MODE_MESSAGE_LIST	2	Received message list display mode
SCE_NP_MESSAGE_DIALOG_MODE_MESSAGE_DETAIL	3	Received message details display mode
SCE_NP_MESSAGE_DIALOG_MODE_MESSAGE_GET	4	Message data obtaining mode

Specify the message type in *type*. One of the following values is specified.

Value	(Number)	Description
SCE_NP_MESSAGE_DIALOG_TYPE_APP_DATA_ATTACHMENT	1	Message with custom data attachments
SCE_NP_MESSAGE_DIALOG_TYPE_INVITATION_ATTACHMENT	2	Message with invitation data attachments
SCE_NP_MESSAGE_DIALOG_TYPE_STORE_LINK_ATTACHMENT	3	Message with store link attachments

Specify dialog option in *options*. The following value is specified.

When not specifying dialog option, specify 0.

SCE_NP_MESSAGE_DIALOG_OPTION_FORBID_EDIT_MESSAGE and SCE_NP_MESSAGE_DIALOG_OPTION_PERMIT_EDIT_PREDEFINED_MESSAGE cannot be specified at the same time. Specify one or the other.

Value	(Number)	Description
SCE_NP_MESSAGE_DIALOG_OPTION_ASSUME_SEND	0x00000001	Sends message without pressing the **Send** button in the message transmission dialog
SCE_NP_MESSAGE_DIALOG_OPTION_FORBID_EDIT_MESSAGE	0x00000002	Disables changing text by users in the message transmission dialog
SCE_NP_MESSAGE_DIALOG_OPTION_SET_USED_FLAG	0x00000003	Sets attached data to used in the message data obtainment dialog
SCE_NP_MESSAGE_DIALOG_OPTION_PERMIT_EDIT_PREDEFINED_MESSAGE	0x00000004	Allows the content of the message specified for *sendParam.body* with the message transmission dialog to be edited by users

Any arbitrary data on the application side can be specified in *userData*. This value is stored in the SceNpMessageDialogResult structure, which can be obtained with sceNpMessageDialogGetResult().

Either of *sendParam*, *listParam*, *detailParam*, or *getParam* must be set according to *mode*.

If SCE_NP_MESSAGE_DIALOG_MODE_MESSAGE_SEND was specified in *mode*, specify the structure that stores the message transmission dialog parameter in *sendParam*. For details, refer to the SceNpMessageDialogMessageSendParam section. If another operation mode was specified, NULL must be specified in *sendParam*.

If SCE_NP_MESSAGE_DIALOG_MODE_MESSAGE_LIST was specified in *mode*, specify the structure that stores the message list dialog parameter in *listParam*. For details, refer to the SceNpMessageDialogMessageListParam section. If another operation mode was specified, NULL must be specified in *listParam*.

If SCE_NP_MESSAGE_DIALOG_MODE_MESSAGE_DETAIL was specified in *mode*, pass the structure that stores the message details dialog parameter in *detailParam*. For details, refer to the SceNpMessageDialogMessageDetailParam section. If another operation mode was specified, NULL must be specified in *detailParam*.

If SCE_NP_MESSAGE_DIALOG_MODE_MESSAGE_GET was specified in *mode*, pass the structure that stores the message data obtainment dialog parameter in *getParam*. For details, refer to the SceNpMessageDialogMessageGetParam section. If another operation mode was specified, NULL must be specified in *getParam*.

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

See Also

sceNpMessageDialogInit(), sceNpMessageDialogParamInit(), SceNpMessageDialogMessageSendParam, SceNpMessageDialogMessageListParam, SceNpMessageDialogMessageDetailParam, SceNpMessageDialogMessageGetParam

SceNpMessageDialogMessageSendParam

Message transmission dialog parameter

Definition

```
#include <np_message_dialog.h>
typedef struct SceNpMessageDialogMessageSendParam{
        SceNpId *npIds;
        SceSize npIdsCount;
        SceChar8 *body;
        void *data;
        SceSize dataSize;
        SceChar8 *indexIconPath;
        SceUInt16 expireMinutes;
        SceChar8 reserved2[2];
        SceSize userSelectableNpIdsCount;
        SceChar8 reserved[30];
} SceNpMessageDialogMessageSendParam;
```

Members

npIds	NP ID of recipient
npIdsCount	Number of NP IDs specified for the recipient
body	Message body (NULL terminated, UTF-8)
data	Pointer to extended data
dataSize	Size of extended data set in `data`
indexIconPath	Path of index icon
expireMinutes	Expiration date of custom data and invitation data (in minutes)
reserved2	Reserved area (fill with all 0s)
userSelectableNpIdsCount	Maximum number of recipients that can be specified by users
reserved	Reserved area (fill with all 0s)

Description

This is the structure of the parameter for displaying message transmission dialog.

It is used when calling NP Message Dialog in the `SCE_NP_MESSAGE_DIALOG_MODE_MESSAGE_SEND` mode. First, initialize the structure filled with all 0s and then set values to the required members.

In *npIds*, specify the NP ID of the recipient user. To specify multiple recipients, specify the start address of the `SceNpId` array. To not specify a recipient, specify NULL. The specified NP ID must be held until NP Message Dialog is terminated by calling `sceNpMessageDialogTerm()`.

In *npIds*, the NP ID of the original message sender can be specified, but messages with store link attachments cannot be sent if the NP ID of only the message sender is specified as the recipient of the message. Messages can be sent when the NP ID of another user besides the message sender is input in *npIds*. This limitation does not apply for messages with invitation data attachments and messages with custom data attachments.

In *npIdsCount*, specify the number of recipients specified in *npIds*. If no recipient was specified, specify 0. The maximum number that can be set in *npIdsCount* is defined with `SCE_NP_MESSAGE_MAX_RECIPIENTS`.

In *body*, specify the message body. The character code must be UTF-8, and the character string termination must be NULL. The maximum size is `SCE_NP_MESSAGE_MAX_BODY_SIZE`, which includes the NULL termination. The specified character string must be held until NP Message Dialog is terminated by calling `sceNpMessageDialogTerm()`.

In *data*, specify the pointer to the extended data (if transmitting a message with custom data attachments, the custom data to be attached, if transmitting a message with invitation data attachments, the invitation data to be attached, and if transmitting a message with store link attachments, the store link data to be attached). Always specify extended data. If NULL is specified, "Could not send the message" will be displayed in the error dialog and the transmission will fail. The specified extended data must be held until NP Message Dialog is terminated by calling `sceNpMessageDialogTerm()`.

The following link information can be included in store link data.

Link Operation	Link Format	Parameter	Parameter Contents
Start Title Store application with a product ID specification	psts:browse?product =PRODUCT_ID	PRODUCT_ID	Product ID of a product distributed by the PlayStation®Store (Title Store)
Start Title Store application with a category ID specification	psts:browse?category =CATEGORY_ID	CATEGORY_ID	PlayStation®Store (Title Store) category ID

In *dataSize*, specify the size of the extended data specified in *data*. The maximum size is `SCE_NP_MESSAGE_MAX_ATTACHMENT_SIZE`. Specification of extended data is always required; do not specify 0.

In *indexIconPath*, specify the path of the index icon file. To not specify an index icon, specify NULL. The index icon must be a PNG/JPEG file and the end of the pathname must be ".jpg" or ".png". The maximum size of the index icon file is `SCE_NP_MESSAGE_MAX_INDEX_ICON_SIZE`. The specified path name and file must be held until NP Message Dialog is terminated by calling `sceNpMessageDialogTerm()`. It cannot be specified for a message with store link attachments.

In *expireMinutes*, specify the expiration date of the custom data or invitation data to be attached in minutes. To not specify an expiration date for custom data, specify 0. In the case of invitation data, an expiration date is required; be sure to specify a value other than 0.

In *userSelectableNpIdsCount*, specify the maximum number of recipients that can be specified by users. If 0 or a value larger than `SCE_NP_MESSAGE_MAX_RECIPIENTS` is specified, the maximum number will be `SCE_NP_MESSAGE_MAX_RECIPIENTS`.

reserved and *reserved2* are reserved areas for future functionality expansion. They must be filled with all 0s.

See Also

`sceNpMessageDialogInit()`, `SceNpMessageDialogParam`

SceNpMessageDialogMessageListParam

Received message list dialog parameter

Definition

```
#include <np_message_dialog.h>
typedef struct SceNpMessageDialogMessageListParam {
        SceChar8 reserved[32];
} SceNpMessageDialogMessageListParam;
```

Members

reserved	Reserved area (fill with all 0s)

Description

This is the structure of the parameter for displaying the message list dialog. It is used when calling NP Message Dialog in the `SCE_NP_MESSAGE_DIALOG_MODE_MESSAGE_LIST` mode. First, initialize the structure filled with all 0s and then set values to the required members.

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

See Also

`sceNpMessageDialogInit()`, `SceNpMessageDialogParam`

SceNpMessageDialogMessageDetailParam

Received message details dialog parameter

Definition

```
#include <np_message_dialog.h>
typedef struct SceNpMessageDialogMessageDetailParam {
        SceNpMessageId messageId;
        SceChar8 reserved[32];
} SceNpMessageDialogMessageDetailParam;
```

Members

messageId	Message ID of a message for detailed display
reserved	Reserved area (fill with all 0s)

Description

This is the structure of the parameter for displaying the message details dialog. It is used when calling NP Message Dialog in the `SCE_NP_MESSAGE_DIALOG_MODE_MESSAGE_DETAIL` mode. First, initialize the structure filled with all 0s and then set values to the required members.

Specify the message ID of the message for detailed display in *messageId*.

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

See Also

`sceNpMessageDialogInit()`, `SceNpMessageDialogParam`

SceNpMessageDialogMessageGetParam

Message data obtainment dialog parameter

Definition

```
#include <np_message_dialog.h>
typedef struct SceNpMessageDialogMessageGetParam {
        SceNpMessageId messageId;
        void *messageBody;
        SceSize messageBodySize;
        void *attachedData;
        SceSize attachedDataSize;
        SceChar8 reserved[32];
} SceNpMessageDialogMessageGetParam;
```

Members

messageId	Message ID of a message to be obtained
messageBody	Pointer where to obtain MessageBody
messageBodySize	Memory size allocated to the pointer where to obtain MessageBody
attachedData	Pointer where to obtain AttachedData
attachedDataSize	Memory size allocated to the pointer where to obtain AttachedData
reserved	Reserved area (fill with all 0s)

Description

This is the structure of the parameter for displaying the message data obtainment dialog. It is used when calling NP Message Dialog in the `SCE_NP_MESSAGE_DIALOG_MODE_MESSAGE_GET` mode. First, initialize the structure filled with all 0s and then set values to the required members.

Specify the message ID indicating the message to be obtained in *messageId*.

Specify the pointer storing the MessageBody of the message to be obtained in *messageBody*.
The specified area must be held until NP Message Dialog is terminated by calling `sceNpMessageDialogTerm()`.

In *messageBodySize*, specify the size of the area allocated to the pointer storing the MessageBody of the message to be obtained.

In *attachedData*, specify the pointer storing the AttachedData of the message to be obtained.
The specified area must be held until NP Message Dialog is terminated by calling `sceNpMessageDialogTerm()`.

In *attachedDataSize*, specify the size of the area allocated to the pointer storing the AttachedData of the message to be obtained.

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

See Also

`sceNpMessageDialogInit(),SceNpMessageDialogParam`

Obtaining NP Message Dialog Operation Status

sceNpMessageDialogGetStatus

Get operation status of NP Message Dialog

Definition

```
#include <np_message_dialog.h>
SceCommonDialogStatus sceNpMessageDialogGetStatus()
```

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	0x0	NP Message Dialog is not running
SCE_COMMON_DIALOG_STATUS_RUNNING	0x1	NP Message Dialog is running
SCE_COMMON_DIALOG_STATUS_FINISHED	0x2	NP Message Dialog operation has finished

Description

This function gets the operation status of NP Message Dialog.

The default value of the operation status is SCE_COMMON_DIALOG_STATUS_NONE.

When calling sceNpMessageDialogInit() is successful, the operation status immediately changes to SCE_COMMON_DIALOG_STATUS_RUNNING.

When NP Message Dialog is closed through user action or sceNpMessageDialogAbort(), the operation status changes to SCE_COMMON_DIALOG_STATUS_FINISHED within a moment.

sceNpMessageDialogGetResult() and sceNpMessageDialogTerm() can be called only while the operation status is SCE_COMMON_DIALOG_STATUS_FINISHED.

When sceNpMessageDialogTerm() is called, the operation status immediately changes to SCE_COMMON_DIALOG_STATUS_NONE.

This function is multithread safe.

See Also

sceNpMessageDialogInit(), sceNpMessageDialogAbort(), sceNpMessageDialogGetResult(), sceNpMessageDialogTerm()

Aborting NP Message Dialog

sceNpMessageDialogAbort

Abort call of NP Message Dialog

Definition

```
#include <np_message_dialog.h>
SceInt32 sceNpMessageDialogAbort()
```

Arguments

None

Return Values

Returns SCE_OK(0) as the value of the function for success.

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

Value	(Number)	Description
SCE_COMMON_DIALOG_ERROR_NOT_IN_USE	0x80020411	sceNpMessageDialogInit() is not called

Description

This function aborts calling of NP Message Dialog. Call this function to promptly abort Message Dialog display, for example when an urgent interrupt must be processed.

It can be called at any time between when sceNpMessageDialogInit() is called and sceNpMessageDialogTerm() 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 dialog being displayed, 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 sceNpMessageDialogGetStatus() section.

When NP Message Dialog is closed with this function, calling sceNpMessageDialogGetResult() returns the following.

SceNpMessageDialogResult.*result* : SCE_COMMON_DIALOG_RESULT_ABORTED

This function is multithread safe.

Examples

```
SceCommonDialogStatus stat;

while(1) {
        stat = sceNpMessageDialogGetStatus();
        if( stat == SCE_COMMON_DIALOG_STATUS_RUNNING ) {
              if( need_abort ) {
                     sceNpMessageDialogAbort();
                     break;
              }
        }
        else if( stat == SCE_COMMON_DIALOG_STATUS_FINISHED ) {
              sceNpMessageDialogTerm();
              break;
        }
}
```

See Also

```
sceNpMessageDialogGetResult()
```

Obtaining NP Message Dialog Call Result

sceNpMessageDialogGetResult

Get call result of NP Message Dialog

Definition

```
#include <np_message_dialog.h>
SceInt32 sceNpMessageDialogGetResult(
        SceNpMessageDialogResult *result
)
```

Arguments

`result` Stores the call result

Return Values

Returns `SCE_OK(0)` as the value of the function for success.

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

Value	(Number)	Description
`SCE_COMMON_DIALOG_ERROR_NULL`	0x80020402	NULL was specified in the argument *result*
`SCE_COMMON_DIALOG_ERROR_NOT_FINISHED`	0x80020410	Called during other than the appropriate operation status (details below)

Description

This function obtains the call result of NP Message Dialog.

This function can be called only while the operation status of NP Message Dialog is `SCE_COMMON_DIALOG_STATUS_FINISHED`. If it is called at times other than the above, `SCE_COMMON_DIALOG_ERROR_NOT_FINISHED` is returned. For details on the operation statuses, refer to the `sceNpMessageDialogGetStatus()` section.

The call result of Message Dialog is stored in the `SceNpMessageDialogResult` structure indicated by *result*. For details on the call results, refer to the `SceNpMessageDialogResult` section.

Be sure to initialize the `SceNpMessageDialogResult` structure indicated by *result* before passing it to this function.

This function is multithread safe.

See Also

`SceNpMessageDialogResult`, `sceNpMessageDialogGetStatus()`

SceNpMessageDialogResult

Structure for obtaining NP Message Dialog call result

Definition

```
#include <np_message_dialog.h>
typedef struct SceNpMessageDialogResult {
        SceNpMessageDialogMode mode;
        SceNpMessageDialogType type;
        void *userData;
        SceNpMessageEntry messageEntry;
        SceNpMessageDialogResultUserAction userAction;
        SceInt32 result;
        SceSize sendNpIdCount;
        SceNpId *sendNpIds;
        SceSize messageBodySize;
        SceSize attachedDataSize;
        SceChar8 reserved[16];
} SceNpMessageDialogResult;
```

Members

mode	*mode* specified at the time of dialog call
type	*type* specified at the time of dialog call
userData	*userData* specified at the time of dialog call
messageEntry	Information of the message selected by the user (when *userAction* is SCE_NP_MESSAGE_DIALOG_USER_ACTION_OPEN_ATTACHED_DATA or SCE_NP_MESSAGE_DIALOG_USER_ACTION_MESSAGE_GET_OK, details below)
userAction	User action that causes Dialog end (details below)
result	Call result (details below)
sendNpIdCount	Number of recipients to which actual message transmission was performed as the result of transmission processing (only when SCE_NP_MESSAGE_DIALOG_USER_ACTION_MESSAGE_SEND_OK is stored in *userAction*, details below)
sendNpIds	Pointer to the SceNpId array for storing the recipients to which message transmission was actually performed as the result of transmission processing (only when SCE_NP_MESSAGE_DIALOG_USER_ACTION_MESSAGE_SEND_OK is stored in *userAction*, details below)
messageBodySize	Size of the MessageBody actually obtained as a result of message obtainment
attachedDataSize	Size of the AttachedData actually obtained as a result of message obtainment
reserved	Reserved area (fill with all 0s)

Description

This structure receives the NP Message Dialog call result. Fill it with all 0s for initialization and pass it to `sceNpMessageDialogGetResult()`.

The value of `SceNpMessageDialogParam.`*`mode`* specified at the call of `sceNpMessageDialogInit()` is stored in *mode*. It can be used to determine which operation mode's calling result it is.

The value of `SceNpMessageDialogParam.`*`type`* specified at the call of `sceNpMessageDialogInit()` is stored in *type*. It can be used to determine which message type's calling result it is.

The value of SceNpMessageDialogParam.*userData* specified at the call of sceNpMessageDialogInit() is stored in *userData*.

When the *userAction* value is SCE_NP_MESSAGE_DIALOG_USER_ACTION_OPEN_ATTACHED_DATA or SCE_NP_MESSAGE_DIALOG_USER_ACTION_MESSAGE_GET_OK, the following message information is stored in *messageEntry*.

messageEntry.messageId	Message ID
messageEntry.dateTime	Date and time the message was sent (local time of PlayStation®Vita on receiving side)
messageEntry.from	Message sender (NP ID)
messageEntry.preview	Message body
messageEntry.expiration_date	Message expiration date
messageEntry.reserved	Reserved area

One of the following values is stored in *userAction*.

Value	(Number)	Description
SCE_NP_MESSAGE_DIALOG_USER_ACTION_MESSAGE_SEND_OK	0x0	Message transmission was successful
SCE_NP_MESSAGE_DIALOG_USER_ACTION_MESSAGE_SEND_NG	0x1	Message transmission failed
SCE_NP_MESSAGE_DIALOG_USER_ACTION_OPEN_ATTACHED_DATA	0x2	Attached data was executed
SCE_NP_MESSAGE_DIALOG_USER_ACTION_CANCEL	0x3	Canceled operation
SCE_NP_MESSAGE_DIALOG_USER_ACTION_MESSAGE_GET_OK	0x4	Message obtainment successful
SCE_NP_MESSAGE_DIALOG_USER_ACTION_MESSAGE_GET_NG	0x5	Message obtainment failed

The call result of NP Message Dialog is stored in *result*. In the case of success, one of the following positive values is stored.

Value	(Number)	Description
SCE_COMMON_DIALOG_RESULT_OK	0x0	User performed **OK** operation.
SCE_COMMON_DIALOG_RESULT_USER_CANCELED	0x1	User performed cancel operation.
SCE_COMMON_DIALOG_RESULT_ABORTED	0x2	Aborted with sceNpMessageDialogAbort()

When the transmission processing is successful (SCE_NP_MESSAGE_DIALOG_USER_ACTION_MESSAGE_SEND_OK is stored in *userAction*), setting the pointer to the array for which SceNpId has been allocated for the number of SCE_NP_MESSAGE_MAX_RECIPIENTS for *sendNpIds* at the call of sceNpMessageDialogGetResult(), the NP IDs of the users that were actually used as recipients are stored in *sendNpId*.

The number of recipients for which transmission processing was performed is stored in *sendNpIdCount*.

If the NP IDs of the recipients for which transmission processing was performed are not required, specify NULL in *sendNpIds* and call sceNpMessageDialogGetResult().

If SCE_NP_MESSAGE_DIALOG_MODE_MESSAGE_GET is specified in *mode*, the size of the MessageBody of the message that was actually obtained will be stored in *messageBodySize*.

If SCE_NP_MESSAGE_DIALOG_MODE_MESSAGE_GET is specified in *mode*, the size of the AttachedData of the message that was actually obtained will be stored in *attachedDataSize*.

reserved is a reserved area for future function expansion. Currently, this area is filled with all 0s.

See Also

`sceNpMessageDialogInit()`, `sceNpMessageDialogAbort()`, `sceNpMessageDialogGetResult()`

End NP Message Dialog

sceNpMessageDialogTerm

End NP Message Dialog

Definition

```
#include <np_message_dialog.h>
SceInt32 sceNpMessageDialogTerm()
```

Arguments

None

Return Values

Returns `SCE_OK(0)` as the value of the function for success.

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

Value	(Number)	Description
SCE_COMMON_DIALOG_ERROR_NOT_FINISHED	0x80020410	Called during other than the appropriate operation status (details below)
SCE_COMMON_DIALOG_ERROR_NOT_IN_USE	0x80020411	sceNpMessageDialogInit() is not called

Description

This function ends calling of NP Message Dialog. NP Message Dialog must be ended with this function after it has been called with `sceNpMessageDialogInit()`.

This function can be called only while the operation status of NP Message Dialog is `SCE_COMMON_DIALOG_STATUS_FINISHED`. If it is called during times other than the above, `SCE_COMMON_DIALOG_ERROR_NOT_FINISHED` is returned.
`SCE_COMMON_DIALOG_ERROR_NOT_IN_USE` will be returned if the NP Message 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 `sceNpMessageDialogGetStatus()` section.

This function is multithread safe.

Examples

```
SceNpCommonDialogStatus stat;

while(1) {
        stat = sceNpMessageDialogGetStatus();
        if( stat == SCE_COMMON_DIALOG_STATUS_FINISHED ) {
              sceNpMessageDialogTerm();
              break;
        }
}
```

See Also

`sceNpMessageDialogInit(),sceNpMessageDialogGetStatus()`

Initialization Functions of NP Message Library

sceNpMessageParamInit

Macro for call parameter initialization

Definition

```
#include <np/np_message.h>
static inline
void sceNpMessageParamInit(SceNpMessageParam *param)
{
        sceClibMemset( param, 0x0, sizeof(SceNpMessageParam) );
        param->sdkVersion = SCE_PSP2_SDK_VERSION;
        param->messageType = SCE_NP_MESSAGE_TYPE_INVALID;
}
```

Arguments

param Call parameter to be initialized

Return Values

None

Description

This is a macro function for initializing the NP Message call parameter.

Examples

```
// Initialize the parameter structure with sceNpMessageParamInit()
SceNpMessageParam    param;
sceNpMessageParamInit(&param);
```

See Also

`SceNpMessageParam`

sceNpMessageInitWithParam

Initialize NP Message library with parameters

Definition

```
#include <np/np_message.h>
SceInt32
sceNpMessageInitWithParam(
        const SceNpMessageParam *param
        );
```

Calling Conditions

Not multithread safe.

Arguments

param Call parameter

Return Values

Returns 0 for normal termination.

Returns a negative value upon error. Main error codes are indicated below; however, program your application so that it does not malfunction when other errors return.

Value	(Number)	Description
SCE_NP_MESSAGE_ERROR_ALREADY_INITIALIZED	0x80551e01	Already initialized sceNpMessageInit() may have already been executed. Check the calling order.
SCE_NP_MESSAGE_ERROR_INVALID_ARGUMENT	0x80551e03	Invalid argument An argument required to execute the function may not have been specified. Check the argument value.
SCE_NP_MESSAGE_ERROR_OUT_OF_MEMORY	0x80551e05	Not enough memory
SCE_NP_MESSAGE_ERROR_COULD_NOT_CONNECT_TO_SERVER	0x80551e06	Failed to connect to the server
SCE_NP_MESSAGE_ERROR_INTERNAL	0x80551e07	Internal error within library
SCE_NP_MESSAGE_ERROR_INVALID_NPCOMMID	0x80551e08	The NP_COMMUNICATION_ID is invalid (param.sfo). NP_COMMUNICATION_ID in the param.sfo may not have been specified correctly. Check param.sfo.
SCE_NP_MESSAGE_FATAL_DB_ERROR	0x80551e0a	Database anomaly
SCE_NP_MESSAGE_ERROR_NOT_SIGNIN	0x80551e0c	Not signed in
SCE_NP_MESSAGE_ERROR_INTERNAL_CACHE	0x80551e0d	Cache operation error (internal error within library)

For other error codes returned by the NP Message library, refer to each applicable reference document.

Description

This function initializes the NP Message library with parameters.

Examples

```
// Initialize the call parameter SceNpMessageParam structure
SceNpMessageParam    param;
sceNpMessageParamInit(&param);

// Set call parameters
param.messageType      = SCE_NP_MESSAGE_TYPE_INVITATION_ATTACHMENT;
param.filter           = SCE_NP_MESSAGE_FILTER_INCLUDE_USED |
                         SCE_NP_MESSAGE_FILTER_INCLUDE_EXPIRED;

// Initialize the NP Message library
ret = sceNpMessageInitWithParam(&param)
if (ret < 0) {
        // Error handling
}
```

See Also

SceNpMessageParam, sceNpMessageTerm()

SceNpMessageParam

NP message library call parameter

Definition

```
#include <np/np_message.h>
typedef struct SceNpMessageParam {
        SceUInt32 sdkVersion;
        SceNpMessageType messageType;
        SceNpMessageFilter filter;
        SceChar8 reserved[32];
} SceNpMessageParam;
```

Members

sdkVersion	SDK version
messageType	Message type
filter	Filter
reserved	Reserved area (fill with all 0s)

Description

This is a structure passed to `sceNpMessageInitWithParam()` to initialize NP Message library. First, use `sceNpMessageParamInit()` to initialize it and set a value to each member.

Specify the SDK version in *sdkVersion*. Since an appropriate value is input when the structure is initialized with `sceNpMessageParamInit()`, do not change the value.

Specify the type of message for processing in *messageType*. One of the following values must be specified.

Value	(Number)	Description
SCE_NP_MESSAGE_TYPE_APP_DATA_ATTACHMENT	1	Custom data attached message type
SCE_NP_MESSAGE_TYPE_INVITATION_ATTACHMENT	2	Invitation data attached message type
SCE_NP_MESSAGE_TYPE_STORE_LINK_ATTACHMENT	3	Store link attached message type

In *filter*, specify the attributes of the messages included in the reception list results when performing message list acquisition. Multiple specifications by ORing the values in the following table are possible.

Value	(Number)	Description
SCE_NP_MESSAGE_FILTER_NONE	0x0000	Among the messages that are sent to this application, only those that are unused and within the expiration date are included in this list. Normally this is the mode that is used.
SCE_NP_MESSAGE_FILTER_INCLUDE_USED	0x0001	In addition to `SCE_NP_MESSAGE_FILTER_NONE`, the messages that have been set as used with the `sceNpMessageSetAttachedDataUsedFlag()` function are included in the list.
SCE_NP_MESSAGE_FILTER_INCLUDE_EXPIRED	0x0002	In addition to `SCE_NP_MESSAGE_FILTER_NONE`, the messages whose expiration date has expired are also included in the list.

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

See Also

`sceNpMessageInitWithParam()`, `sceNpMessageParamInit()`

sceNpMessageInit

Initialize the NP Message library

Definition

```
#include <np/np_message.h>
SceInt32
sceNpMessageInit(
        const SceNpMessageType messageType
        );
```

Calling Conditions

Not multithread safe.

Arguments

messageType	Message type

Return Values

Returns 0 for normal termination.

Returns a negative value upon error. Main error codes are indicated below; however, program your application so that it does not malfunction when other errors return.

Value	(Number)	Description
SCE_NP_MESSAGE_ERROR_ALREADY_INITIALIZED	0x80551e01	Already initialized sceNpMessageInit() may have already been executed. Check the calling order.
SCE_NP_MESSAGE_ERROR_INVALID_ARGUMENT	0x80551e03	Invalid argument An argument required to execute the function may not have been specified. Check the argument value.
SCE_NP_MESSAGE_ERROR_OUT_OF_MEMORY	0x80551e05	Not enough memory
SCE_NP_MESSAGE_ERROR_COULD_NOT_CONNECT_TO_SERVER	0x80551e06	Failed to connect to the server
SCE_NP_MESSAGE_ERROR_INTERNAL	0x80551e07	Internal error within library
SCE_NP_MESSAGE_ERROR_INVALID_NPCOMMID	0x80551e08	The NP_COMMUNICATION_ID is invalid (param.sfo). NP_COMMUNICATION_ID in the param.sfo may not have been specified correctly. Check param.sfo.
SCE_NP_MESSAGE_FATAL_DB_ERROR	0x80551e0a	Database anomaly
SCE_NP_MESSAGE_ERROR_NOT_SIGNIN	0x80551e0c	Not signed in
SCE_NP_MESSAGE_ERROR_INTERNAL_CACHE	0x80551e0d	Cache operation error (internal error within library)

For other error codes returned by the NP Message library, refer to each applicable reference document.

Description

This function initializes the NP Message library.

See Also

`sceNpMessageTerm()`

NP Message Library Operating Functions

sceNpMessageSyncMessage

Synchronize messages

Definition

```
#include <np/np_message.h>
SceInt32
sceNpMessageSyncMessage(
        void
        );
```

Calling Conditions

Not multithread safe.

Arguments

None

Return Values

Returns 0 for normal termination.

Returns a negative value upon error. Main error codes are indicated below; however, program your application so that it does not malfunction when other errors return.

Value	(Number)	Description
SCE_NP_MESSAGE_ERROR_NOT_INITIALIZED	0x80551e02	The NP Message library is not initialized sceNpMessageInit() must be executed before the execution of this function.
SCE_NP_MESSAGE_ERROR_INVALID_ARGUMENT	0x80551e03	Invalid argument An argument required to execute the function may not have been specified. Check the argument value.
SCE_NP_MESSAGE_ERROR_ABORTED	0x80551e04	Processing was aborted
SCE_NP_MESSAGE_ERROR_OUT_OF_MEMORY	0x80551e05	Not enough memory
SCE_NP_MESSAGE_ERROR_COULD_NOT_CONNECT_TO_SERVER	0x80551e06	Failed to connect to the server
SCE_NP_MESSAGE_ERROR_INTERNAL	0x80551e07	Internal error within library
SCE_NP_MESSAGE_ERROR_SESSION_INVALID	0x80551e09	The session has become invalid
SCE_NP_MESSAGE_FATAL_DB_ERROR	0x80551e0a	Database anomaly
SCE_NP_MESSAGE_ERROR_INTERNAL_IMAP	0x80551e0b	IMAP communication processing error (internal error within library) This error does not occur with SDK3.000 or later.
SCE_NP_MESSAGE_ERROR_NOT_SIGNIN	0x80551e0c	Not signed in
SCE_NP_MESSAGE_ERROR_INTERNAL_CACHE	0x80551e0d	Cache operation error (internal error within library)
SCE_NP_MESSAGE_ERROR_INTERNAL_MIME	0x80551e0e	MIME processing error (internal error within library)

For other error codes returned by the NP Message library, refer to each applicable reference document.

Description

This function performs synchronization with the message server of PSN℠. If messages that have not been received yet arrive to the message server of PSN℠, they can be received on the PlayStation®Vita.

See Also

```
sceNpMessageAbort()
```

sceNpMessageGetMessageEntryCount

Get number of messages

Definition

```
#include <np/np_message.h>
SceInt32
sceNpMessageGetMessageEntryCount(
        SceInt32 *totalCount,
        SceInt32 *newCount
        );
```

Calling Conditions

Not multithread safe.

Arguments

totalCount	Variable for storing the number of messages
newCount	Variable for storing the number of unread messages

Return Values

Returns 0 for normal termination.

Returns a negative value upon error. Main error codes are indicated below; however, program your application so that it does not malfunction when other errors return.

Value	(Number)	Description
SCE_NP_MESSAGE_ERROR_NOT_INITIALIZED	0x80551e02	The NP Message library is not initialized sceNpMessageInit() must be executed before the execution of this function.
SCE_NP_MESSAGE_ERROR_INVALID_ARGUMENT	0x80551e03	Invalid argument An argument required to execute the function may not have been specified. Check the argument value.
SCE_NP_MESSAGE_ERROR_ABORTED	0x80551e04	Processing was aborted
SCE_NP_MESSAGE_ERROR_OUT_OF_MEMORY	0x80551e05	Not enough memory
SCE_NP_MESSAGE_ERROR_COULD_NOT_CONNECT_TO_SERVER	0x80551e06	Failed to connect to the server
SCE_NP_MESSAGE_ERROR_INTERNAL	0x80551e07	Internal error within library
SCE_NP_MESSAGE_ERROR_SESSION_INVALID	0x80551e09	The session has become invalid
SCE_NP_MESSAGE_FATAL_DB_ERROR	0x80551e0a	Database anomaly
SCE_NP_MESSAGE_ERROR_INTERNAL_IMAP	0x80551e0b	IMAP communication processing error (internal error within library) This error does not occur with SDK3.000 or later.
SCE_NP_MESSAGE_ERROR_NOT_SIGNIN	0x80551e0c	Not signed in
SCE_NP_MESSAGE_ERROR_INTERNAL_CACHE	0x80551e0d	Cache operation error (internal error within library)
SCE_NP_MESSAGE_ERROR_INTERNAL_MIME	0x80551e0e	MIME processing error (internal error within library)

For other error codes returned by the NP Message library, refer to each applicable reference document.

Description

This function obtains the total number of messages and the number of unread messages.

See Also

`sceNpMessageAbort()`

sceNpMessageGetMessageEntries

Get list of messages

Definition

```
#include <np/np_message.h>
SceInt32
sceNpMessageGetMessageEntries(
        SceInt32 offset,
        SceInt32 limit,
        SceNpMessageEntry *entries,
        SceInt32 *storedCount
        );
```

Calling Conditions

Not multithread safe.

Arguments

offset	Offset
limit	Limit (1 to 50 can be specified)
entries	List of messages
storedCount	Actual number of messages obtained

Return Values

Returns 0 for normal termination.

Returns a negative value upon error. Main error codes are indicated below; however, program your application so that it does not malfunction when other errors return.

Value	(Number)	Description
SCE_NP_MESSAGE_ERROR_NOT_INITIALIZED	0x80551e02	The NP Message library is not initialized sceNpMessageInit() must be executed before the execution of this function.
SCE_NP_MESSAGE_ERROR_INVALID_ARGUMENT	0x80551e03	Invalid argument An argument required to execute the function may not have been specified. Check the argument value.
SCE_NP_MESSAGE_ERROR_ABORTED	0x80551e04	Processing was aborted
SCE_NP_MESSAGE_ERROR_OUT_OF_MEMORY	0x80551e05	Not enough memory
SCE_NP_MESSAGE_ERROR_COULD_NOT_CONNECT_TO_SERVER	0x80551e06	Failed to connect to the server
SCE_NP_MESSAGE_ERROR_INTERNAL	0x80551e07	Internal error within library
SCE_NP_MESSAGE_ERROR_SESSION_INVALID	0x80551e09	The session has become invalid
SCE_NP_MESSAGE_FATAL_DB_ERROR	0x80551e0a	Database anomaly
SCE_NP_MESSAGE_ERROR_INTERNAL_IMAP	0x80551e0b	IMAP communication processing error (internal error within library) This error does not occur with SDK3.000 or later.
SCE_NP_MESSAGE_ERROR_NOT_SIGNIN	0x80551e0c	Not signed in

Value	(Number)	Description
`SCE_NP_MESSAGE_ERROR_INTERNAL_CACHE`	0x80551e0d	Cache operation error (internal error within library)
`SCE_NP_MESSAGE_ERROR_INTERNAL_MIME`	0x80551e0e	MIME processing error (internal error within library)

For other error codes returned by the NP Message library, refer to each applicable reference document.

Description

This function obtains entry information for messages up to the maximum specified in *`limit`*, in order from the newest message received starting from the message specified in *`offset`*. Obtained results are stored in *`entries`*. In order to store the number of entries specified in *`limit`*, it is necessary to set the pointer to the array of the `SceNpMessageEntry` structure in advance before calling this function.

The number of entries that was actually obtained successfully is stored in *`storedCount`*.

See Also

`SceNpMessageEntry`, `sceNpMessageAbort()`

sceNpMessageGetMessageEntry

Get message entry

Definition

```
#include <np/np_message.h>
SceInt32
sceNpMessageGetMessageEntry(
        SceNpMessageId *messageId,
        SceNpMessageEntry *entry
        );
```

Calling Conditions

Not multithread safe.

Arguments

messageId	Message ID
entry	Entry actually obtained

Return Values

Returns 0 for normal termination.

Returns a negative value upon error. Main error codes are indicated below; however, program your application so that it does not malfunction when other errors return.

Value	(Number)	Description
SCE_NP_MESSAGE_ERROR_NOT_INITIALIZED	0x80551e02	The NP Message library is not initialized sceNpMessageInit() must be executed before the execution of this function.
SCE_NP_MESSAGE_ERROR_INVALID_ARGUMENT	0x80551e03	Invalid argument An argument required to execute the function may not have been specified. Check the argument value.
SCE_NP_MESSAGE_ERROR_ABORTED	0x80551e04	Processing was aborted
SCE_NP_MESSAGE_ERROR_OUT_OF_MEMORY	0x80551e05	Not enough memory
SCE_NP_MESSAGE_ERROR_COULD_NOT_CONNECT_TO_SERVER	0x80551e06	Failed to connect to the server
SCE_NP_MESSAGE_ERROR_INTERNAL	0x80551e07	Internal error within library
SCE_NP_MESSAGE_ERROR_SESSION_INVALID	0x80551e09	The session has become invalid
SCE_NP_MESSAGE_FATAL_DB_ERROR	0x80551e0a	Database anomaly
SCE_NP_MESSAGE_ERROR_INTERNAL_IMAP	0x80551e0b	IMAP communication processing error (internal error within library) This error does not occur with SDK3.000 or later.
SCE_NP_MESSAGE_ERROR_NOT_SIGNIN	0x80551e0c	Not signed in
SCE_NP_MESSAGE_ERROR_INTERNAL_CACHE	0x80551e0d	Cache operation error (internal error within library)
SCE_NP_MESSAGE_ERROR_INTERNAL_MIME	0x80551e0e	MIME processing error (internal error within library)

For other error codes returned by the NP Message library, refer to each applicable reference document.

Description

This function obtains the message entry specified with *`messageId`*.
The entry that was actually obtained is stored in *`entry`*.

See Also

`SceNpMessageId`, `SceNpMessageEntry`, `sceNpMessageAbort()`

sceNpMessageGetMessage

Get message body

Definition

```
#include <np/np_message.h>
SceInt32
sceNpMessageGetMessage(
        SceNpMessageId *messageId,
        SceChar8 *buffer,
        SceSize bufferSize,
        SceSize *storedSize
        );
```

Calling Conditions

Not multithread safe.

Arguments

messageId	Message ID
buffer	Buffer to store message body
bufferSize	Size of buffer to store message body
storedSize	Size actually obtained

Return Values

Returns 0 for normal termination.

Returns a negative value upon error. Main error codes are indicated below; however, program your application so that it does not malfunction when other errors return.

Value	(Number)	Description
SCE_NP_MESSAGE_ERROR_NOT_INITIALIZED	0x80551e02	The NP Message library is not initialized sceNpMessageInit() must be executed before the execution of this function.
SCE_NP_MESSAGE_ERROR_INVALID_ARGUMENT	0x80551e03	Invalid argument An argument required to execute the function may not have been specified. Check the argument value.
SCE_NP_MESSAGE_ERROR_ABORTED	0x80551e04	Processing was aborted
SCE_NP_MESSAGE_ERROR_OUT_OF_MEMORY	0x80551e05	Not enough memory
SCE_NP_MESSAGE_ERROR_COULD_NOT_CONNECT_TO_SERVER	0x80551e06	Failed to connect to the server
SCE_NP_MESSAGE_ERROR_INTERNAL	0x80551e07	Internal error within library
SCE_NP_MESSAGE_ERROR_SESSION_INVALID	0x80551e09	The session has become invalid
SCE_NP_MESSAGE_FATAL_DB_ERROR	0x80551e0a	Database anomaly
SCE_NP_MESSAGE_ERROR_INTERNAL_IMAP	0x80551e0b	IMAP communication processing error (internal error within library) This error does not occur with SDK3.000 or later.
SCE_NP_MESSAGE_ERROR_NOT_SIGNIN	0x80551e0c	Not signed in

Value	(Number)	Description
`SCE_NP_MESSAGE_ERROR_INTERNAL_CACHE`	0x80551e0d	Cache operation error (internal error within library)
`SCE_NP_MESSAGE_ERROR_INTERNAL_MIME`	0x80551e0e	MIME processing error (internal error within library)

For other error codes returned by the NP Message library, refer to each applicable reference document.

Description

This function obtains body of a message specified with *`messageId`*.

By calling it specifying the pointer to the buffer storing the message's body in the variable *`buffer`*, and its size in *`bufferSize`*, it is possible to obtain the message's body.

The character code of message characters is UTF-8, while the maximum size is `SCE_NP_MESSAGE_MAX_BODY_SIZE`.

The size of the characters that were actually obtained successfully is stored in *`storedSize`*.

See Also

`SceNpMessageId`, `sceNpMessageAbort()`

sceNpMessageGetAttachedData

Get attached data of message

Definition

```
#include <np/np_message.h>
SceInt32
sceNpMessageGetAttachedData(
        SceNpMessageId *messageId,
        SceChar8 *buffer,
        SceSize bufferSize,
        SceSize *storedSize
        );
```

Calling Conditions

Not multithread safe.

Arguments

messageId	Message ID
buffer	Buffer storing attached data
bufferSize	Size of buffer storing attached data
storedSize	Size actually obtained

Return Values

Returns 0 for normal termination.

Returns a negative value upon error. Main error codes are indicated below; however, program your application so that it does not malfunction when other errors return.

Value	(Number)	Description
SCE_NP_MESSAGE_ERROR_NOT_INITIALIZED	0x80551e02	The NP Message library is not initialized sceNpMessageInit() must be executed before the execution of this function.
SCE_NP_MESSAGE_ERROR_INVALID_ARGUMENT	0x80551e03	Invalid argument An argument required to execute the function may not have been specified. Check the argument value.
SCE_NP_MESSAGE_ERROR_ABORTED	0x80551e04	Processing was aborted
SCE_NP_MESSAGE_ERROR_OUT_OF_MEMORY	0x80551e05	Not enough memory
SCE_NP_MESSAGE_ERROR_COULD_NOT_CONNECT_TO_SERVER	0x80551e06	Failed to connect to the server
SCE_NP_MESSAGE_ERROR_INTERNAL	0x80551e07	Internal error within library
SCE_NP_MESSAGE_ERROR_SESSION_INVALID	0x80551e09	The session has become invalid
SCE_NP_MESSAGE_FATAL_DB_ERROR	0x80551e0a	Database anomaly
SCE_NP_MESSAGE_ERROR_INTERNAL_IMAP	0x80551e0b	IMAP communication processing error (internal error within library) This error does not occur with SDK3.000 or later.
SCE_NP_MESSAGE_ERROR_NOT_SIGNIN	0x80551e0c	Not signed in

Value	(Number)	Description
SCE_NP_MESSAGE_ERROR_INTERNAL_CACHE	0x80551e0d	Cache operation error (internal error within library)
SCE_NP_MESSAGE_ERROR_INTERNAL_MIME	0x80551e0e	MIME processing error (internal error within library)

For other error codes returned by the NP Message library, refer to each applicable reference document.

Description

This function obtains data attached to a message specified with *messageId*.

By calling it specifying the pointer to the buffer storing the attached data in the variable *buffer*, and its size in *bufferSize*, it is possible to obtain the attached data.

The maximum size of the attached data is SCE_NP_MESSAGE_MAX_ATTACHMENT_SIZE.

The size of the attached data that was actually obtained successfully is stored in *storedSize*.

See Also

SceNpMessageId, sceNpMessageAbort()

sceNpMessageSetAttachedDataUsedFlag

Set attached data as used

Definition

```
#include <np/np_message.h>
SceInt32
sceNpMessageSetAttachedDataUsedFlag(
        SceNpMessageId *messageId
        );
```

Calling Conditions

Not multithread safe.

Arguments

messageId	Message ID

Return Values

Returns 0 for normal termination.

Returns a negative value upon error. Main error codes are indicated below; however, program your application so that it does not malfunction when other errors return.

Value	(Number)	Description
SCE_NP_MESSAGE_ERROR_NOT_INITIALIZED	0x80551e02	The NP Message library is not initialized `sceNpMessageInit()` must be executed before the execution of this function.
SCE_NP_MESSAGE_ERROR_INVALID_ARGUMENT	0x80551e03	Invalid argument An argument required to execute the function may not have been specified. Check the argument value.
SCE_NP_MESSAGE_ERROR_ABORTED	0x80551e04	Processing was aborted
SCE_NP_MESSAGE_ERROR_OUT_OF_MEMORY	0x80551e05	Not enough memory
SCE_NP_MESSAGE_ERROR_COULD_NOT_CONNECT_TO_SERVER	0x80551e06	Failed to connect to the server
SCE_NP_MESSAGE_ERROR_INTERNAL	0x80551e07	Internal error within library
SCE_NP_MESSAGE_ERROR_SESSION_INVALID	0x80551e09	The session has become invalid
SCE_NP_MESSAGE_FATAL_DB_ERROR	0x80551e0a	Database anomaly
SCE_NP_MESSAGE_ERROR_INTERNAL_IMAP	0x80551e0b	IMAP communication processing error (internal error within library) This error does not occur with SDK3.000 or later.
SCE_NP_MESSAGE_ERROR_NOT_SIGNIN	0x80551e0c	Not signed in
SCE_NP_MESSAGE_ERROR_INTERNAL_CACHE	0x80551e0d	Cache operation error (internal error within library)
SCE_NP_MESSAGE_ERROR_INTERNAL_MIME	0x80551e0e	MIME processing error (internal error within library)

For other error codes returned by the NP Message library, refer to each applicable reference document.

Description

This function sets attached data as used for message specified to *`messageId`*.

By setting the attached data to used, it is possible to remove it from the message list displayed by NP Message Dialog. Also, it can be excluded from the results of list obtainment obtained with `sceNpMessageGetMessageEntries()`.

See Also

`SceNpMessageId`, `sceNpMessageAbort()`

sceNpMessageAbort

Abort ongoing processing

Definition

```
#include <np/np_message.h>
SceInt32
sce sceNpMessageAbort(
        void
        );
```

Calling Conditions

Not multithread safe.

Arguments

None

Return Values

Returns 0 for normal termination.

Returns a negative value upon error. Main error codes are indicated below; however, program your application so that it does not malfunction when other errors return.

Value	(Number)	Description
SCE_NP_MESSAGE_ERROR_NOT_INITIALIZED	0x80551e02	The NP Message library is not initialized sceNpMessageInit() must be executed before the execution of this function.
SCE_NP_MESSAGE_ERROR_OUT_OF_MEMORY	0x80551e05	Not enough memory
SCE_NP_MESSAGE_ERROR_COULD_NOT_CONNECT_TO_SERVER	0x80551e06	Failed to connect to the server
SCE_NP_MESSAGE_ERROR_INTERNAL	0x80551e07	Internal error within library
SCE_NP_MESSAGE_ERROR_SESSION_INVALID	0x80551e09	The session has become invalid

For other error codes returned by the NP Message library, refer to each applicable reference document.

Description

This function aborts NP Message library's ongoing processing.

If you wish to use the NP Message library again after executing sceNpMessageAbort(), it will be necessary to call the initialization function sceNpMessageInit() or sceNpMessageInitWithParam() once again.

See Also

sceNpMessageSyncMessage(), sceNpMessageGetMessageEntryCount(), sceNpMessageGetMessageEntries(), sceNpMessageGetMessage(), sceNpMessageGetAttachedData(), sceNpMessageSetAttachedDataUsedFlag()

Termination Functions of NP Message Library

sceNpMessageTerm

Terminate the NP Message library

Definition

```
#include <np/np_message.h>
SceInt32
sceNpMessageTerm(
        void
        );
```

Calling Conditions

Not multithread safe.

Arguments

None

Return Values

Returns 0 for normal termination.

Returns a negative value upon error. Main error codes are indicated below; however, program your application so that it does not malfunction when other errors return.

Value	(Number)	Description
SCE_NP_MESSAGE_ERROR_NOT_INITIALIZED	0x80551e02	The NP Message library is not initialized sceNpMessageInit() must be executed before the execution of this function.
SCE_NP_MESSAGE_ERROR_ABORTED	0x80551e04	Processing was aborted
SCE_NP_MESSAGE_ERROR_OUT_OF_MEMORY	0x80551e05	Not enough memory
SCE_NP_MESSAGE_ERROR_COULD_NOT_CONNECT_TO_SERVER	0x80551e06	Failed to connect to the server
SCE_NP_MESSAGE_ERROR_INTERNAL	0x80551e07	Internal error within library
SCE_NP_MESSAGE_FATAL_DB_ERROR	0x80551e0a	Database anomaly

For other error codes returned by the NP Message library, refer to each applicable reference document.

Description

This function terminates the NP Message library.

See Also

sceNpMessageInit()

Other Definition Value

Maximum Value Definition

Maximum value definition of NP Message Dialog/NP Message library

Definition

Value	(Number)	Description
SCE_NP_MESSAGE_MAX_ RECIPIENTS	16	Maximum number when specifying multiple recipients. Multiple recipients can be specified when transmitting a message in NP Message Dialog. Specify a number of recipients not exceeding the maximum number.
SCE_NP_MESSAGE_MAX_ BODY_SIZE	512	Maximum size of the message body. This is the maximum size of the character string that can be specified as the body of the message to be transmitted with NP Message Dialog. Specify the size of the body, including the NULL termination character, so as not to exceed this maximum size. Also, specify all character strings in UTF-8.
SCE_NP_MESSAGE_MAX_ ATTACHMENT_SIZE	1024* 1024	Maximum size of attached data of game boot message. The maximum size (byte) of the data that can be attached to the message to be transmitted with NP Message Dialog. Specify the data so as not to exceed this maximum size.
SCE_NP_MESSAGE_MAX_ INDEX_ICON_SIZE	64* 1024	Maximum size of index icon image file. An index icon image can be specified when transmitting a message with NP Message Dialog. The maximum size of the index icon image file that can be specified is fixed, so specify an index icon image that does not exceed this maximum size.

List of Error Codes of the NP Message Dialog

List of error codes returned by the NP Message Dialog

Definition

Value	(Number)	Description
`SCE_COMMON_DIALOG_ERROR_BUSY`	0x80020401	The Common Dialog feature is already being called
`SCE_COMMON_DIALOG_ERROR_NULL`	0x80020402	NULL was specified for the *`param`* argument
`SCE_COMMON_DIALOG_ERROR_INVALID_ARGUMENT`	0x80020403	The content of the *`param`* argument is invalid
`SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL`	0x8002047F	Internal error
`SCE_COMMON_DIALOG_ERROR_NOT_IN_USE`	0x80020411	`sceNpMessageDialogInit()` is not called
`SCE_COMMON_DIALOG_ERROR_NULL`	0x80020402	NULL was specified in the argument *`result`*
`SCE_COMMON_DIALOG_ERROR_NOT_FINISHED`	0x80020410	Called during other than the appropriate operation status

List of Error Codes of the NP Message Library

List of error codes returned by the NP Message library

Definition

Value	(Number)	Description
SCE_NP_MESSAGE_ERROR_ALREADY_INITIALIZED	0x80551e01	Already initialized
SCE_NP_MESSAGE_ERROR_NOT_INITIALIZED	0x80551e02	Not initialized
SCE_NP_MESSAGE_ERROR_INVALID_ARGUMENT	0x80551e03	Invalid argument
SCE_NP_MESSAGE_ERROR_ABORTED	0x80551e04	Processing was aborted
SCE_NP_MESSAGE_ERROR_OUT_OF_MEMORY	0x80551e05	Not enough memory
SCE_NP_MESSAGE_ERROR_COULD_NOT_CONNECT_TO_SERVER	0x80551e06	Failed to connect to the server
SCE_NP_MESSAGE_ERROR_INTERNAL	0x80551e07	Internal error within library
SCE_NP_MESSAGE_ERROR_INVALID_NPCOMMID	0x80551e08	The NP_COMMUNICATION_ID is invalid (param.sfo)
SCE_NP_MESSAGE_ERROR_SESSION_INVALID	0x80551e09	The session has become invalid
SCE_NP_MESSAGE_FATAL_DB_ERROR	0x80551e0a	Database anomaly
SCE_NP_MESSAGE_ERROR_INTERNAL_IMAP	0x80551e0b	IMAP communication processing error (internal error within library) This error does not occur with SDK3.000 or later.
SCE_NP_MESSAGE_ERROR_NOT_SIGNIN	0x80551e0c	Not signed in
SCE_NP_MESSAGE_ERROR_INTERNAL_CACHE	0x80551e0d	Cache operation error (internal error within library)
SCE_NP_MESSAGE_ERROR_INTERNAL_MIME	0x80551e0e	MIME processing error (internal error within library)

*In the case that SCE_NP_MESSAGE_ERROR_SESSION_INVALID(0x80551e09) returns, there is a chance of successful completion if a retry is attempted after a duration of certain time.