

NP SNS Facebook Library Reference

© 2014 Sony Computer Entertainment Inc.
All Rights Reserved.
SCE Confidential

Table of Contents

Library Initialization / Termination	3
sceNpSnsFacebookInit	4
sceNpSnsFacebookTerm	5
Request Operations	6
sceNpSnsFacebookCreateRequest	7
sceNpSnsFacebookDeleteRequest	8
sceNpSnsFacebookAbortRequest	9
Communication Processing	10
sceNpSnsFacebookGetLongAccessToken	11
Facebook Permission Dialog Functions	14
sceNpSnsFacebookDialogParamInit	15
sceNpSnsFacebookDialogInit	16
sceNpSnsFacebookDialogTerm	18
sceNpSnsFacebookDialogGetStatus	19
sceNpSnsFacebookDialogGetResultLongToken	20
sceNpSnsFacebookDialogAbort	21
Structures	22
SceNpSnsFacebookAccessTokenParam	23
SceNpSnsFacebookLongAccessTokenResult	24
SceNpSnsFacebookDialogParam	25
SceNpSnsFacebookDialogResultLongToken	26
Constants	27
SCE_NP_SNS_FACEBOOK_MAX_REQUEST_NUM	28
SCE_NP_SNS_FACEBOOK_PERMISSIONS_LENGTH_MAX	29
SCE_NP_SNS_FACEBOOK_LONG_ACCESS_TOKEN_LENGTH_MAX	30
SCE_NP_SNS_FACEBOOK_DIALOG_MODE_LONG_ACCESS_TOKEN	31
Return Codes	32

Library Initialization / Termination

sceNpSnsFacebookInit

Initialize the NP SNS Facebook library

Definition

```
#include <np.h>
SceInt32 sceNpSnsFacebookInit(
    void
);
```

Arguments

None

Return Values

Returns 0 for normal termination.

Returns a negative value upon error. The main error codes are indicated below; however, program your application so that it does not malfunction when an error code other than one of those listed below is returned.

Value	Hexadecimal	Description
SCE_NP_SNS_FACEBOOK_ERROR_ALREADY_INITIALIZED	0x80552602	Already initialized
SCE_NP_SNS_FACEBOOK_ERROR_OUT_OF_MEMORY	0x80552605	Insufficient free memory space

Description

This function initializes the NP SNS Facebook library. Make sure to call this function once before using the NP SNS Facebook library.

Notes

This function is not multithread safe. If it is called at the same time from different threads, the library may later malfunction even if this function terminates normally. Therefore, avoid simultaneous calls when programming.

See Also

sceNpSnsFacebookTerm()

SCE CONFIDENTIAL

sceNpSnsFacebookTerm

Terminate NP SNS Facebook library

Definition

```
#include <np.h>
int sceNpSnsFacebookTerm(
    void
);
```

Arguments

None

Return Values

Returns 0 for normal termination.

Returns a negative value upon error. The main error codes are indicated below; however, program your application so that it does not malfunction when an error code other than one of those listed below is returned.

Value	Hexadecimal	Description
SCE_NP_SNS_FACEBOOK_ERROR_NOT_INITIALIZED	0x80552603	Library has not been initialized

Description

This function terminates the NP SNS Facebook library.

Notes

This function is not multithread safe. If it is called at the same time from different threads, the library may later malfunction even if this function terminates normally. Therefore, avoid simultaneous calls when programming.

See Also

sceNpSnsFacebookInit()

Request Operations

000004892117

SCE CONFIDENTIAL

sceNpSnsFacebookCreateRequest

Create a request of NP SNS Facebook library

Definition

```
#include <np.h>
int sceNpSnsFacebookCreateRequest (
    void
);
```

Arguments

None

Return Values

Returns a request ID (>0) for normal termination.

Returns a negative value upon error. The main error codes are indicated below; however, program your application so that it does not malfunction when an error code other than one of those listed below is returned.

Value	Hexadecimal	Description
SCE_NP_SNS_FACEBOOK_ERROR_NOT_INITIALIZED	0x80552603	Library has not been initialized
SCE_NP_SNS_FACEBOOK_ERROR_OUT_OF_MEMORY	0x80552605	Insufficient free memory space
SCE_NP_SNS_FACEBOOK_ERROR_EXCEEDS_MAX	0x80552606	Created more than 4 requests at one time. (Check to see that sceNpSnsFacebookDeleteRequest () was called as necessary.)

Description

This function creates a request used by the NP SNS Facebook library.

A request ID is an ID that is used up for each communication processing function. A request ID must always be created before using these functions. Moreover, a used up request must be deleted with sceNpSnsFacebookDeleteRequest ().

Up to 4 requests can exist simultaneously.

When sceNpSnsFacebookTerm () is called, all requests will be deleted automatically. However, try to call sceNpSnsFacebookTerm () after delete all the requests explicitly beforehand.

See Also

sceNpSnsFacebookDeleteRequest (), SCE_NP_SNS_FACEBOOK_MAX_REQUEST_NUM

SCE CONFIDENTIAL

sceNpSnsFacebookDeleteRequest

Destroy the request of the NP SNS Facebook library

Definition

```
#include <np.h>
int sceNpSnsFacebookDeleteRequest (
    SceInt32 reqId
);
```

Arguments

reqId Request ID of the NP SNS Facebook library

Return Values

Returns 0 for normal termination.

Returns a negative value upon error. The main error codes are indicated below; however, program your application so that it does not malfunction when an error code other than one of those listed below is returned.

Value	Hexadecimal	Description
SCE_NP_SNS_FACEBOOK_ERROR_NOT_INITIALIZED	0x80552603	Library has not been initialized
SCE_NP_SNS_FACEBOOK_ERROR_OUT_OF_MEMORY	0x80552605	Insufficient free memory space
SCE_NP_SNS_FACEBOOK_ERROR_INVALID_ARGUMENT	0x80552604	ID specified for <i>reqId</i> does not exist

Description

This function destroys the used up request.

See Also

sceNpSnsFacebookCreateRequest ()

SCE CONFIDENTIAL

sceNpSnsFacebookAbortRequest

Abort communication processing

Definition

```
#include <np.h>
int sceNpSnsFacebookAbortRequest (
    SceInt32 reqId
);
```

Arguments

reqId Request ID of the NP SNS Facebook library

Return Values

Returns 0 for normal termination.

Returns a negative value upon error. The main error codes are indicated below; however, program your application so that it does not malfunction when an error code other than one of those listed below is returned.

Value	Hexadecimal	Description
SCE_NP_SNS_FACEBOOK_ERROR_NOT_INITIALIZED	0x80552603	Library has not been initialized
SCE_NP_SNS_FACEBOOK_ERROR_INVALID_ARGUMENT	0x80552604	ID specified for <i>reqId</i> does not exist

Description

This function aborts communication processing.

For *reqId*, specify the ID of the request for the communication processing you wish to abort.

See Also

sceNpSnsFacebookCreateRequest ()

Communication Processing

sceNpSnsFacebookGetLongAccessToken

Obtain an access token for accessing a Facebook® Graph API

Definition

```
#include <np.h>
int sceNpSnsFacebookGetLongAccessToken (
    SceInt32 reqId,
    const SceNpSnsFacebookAccessTokenParam *param,
    SceNpSnsFacebookLongAccessTokenResult *result
);
```

Arguments

reqId Request ID of the NP SNS Facebook library [IN]
param Parameters for obtaining an access token [IN]
 Initialize the structure with 0's first, and then set *fbAppId*, as well as the other required fields.
result Obtained access token and its expiration information [OUT]

Return Values

Returns 0 for normal termination.

Returns a negative value upon error. The main error codes are indicated below; however, program your application so that it does not malfunction when an error code other than one of those listed below is returned.

Value	Hexadecimal	Description
SCE_NP_SNS_FACEBOOK_ERROR_UNKNOWN	0x80552601	An error not listed below occurred (communication error, for example)
SCE_NP_SNS_FACEBOOK_ERROR_NOT_INITIALIZED	0x80552603	Library has not been initialized
SCE_NP_SNS_FACEBOOK_ERROR_INVALID_ARGUMENT	0x80552604	NULL was specified to either <i>Param</i> or <i>result</i>
SCE_NP_SNS_FACEBOOK_ERROR_OUT_OF_MEMORY	0x80552605	Insufficient free memory space
SCE_NP_SNS_FACEBOOK_ERROR_NOT_SIGNIN	0x80552607	Not signed in to PSN SM This function only works when signed in to PSN SM . Call the function in the signed-in state.
SCE_NP_SNS_FACEBOOK_ERROR_CHAT_RESTRICTION	0x80552608	The Facebook coordination features cannot be used due to chat restrictions.
SCE_NP_SNS_FACEBOOK_ERROR_ABORTED	0x80552609	<i>sceNpSnsFacebookAbortRequest()</i> aborted this function's operation. Delete the request.
SCE_NP_SNS_FACEBOOK_ERROR_FACEBOOK_SERVER_ERROR	0x8055260a	Facebook server returned an error A character string pattern that the Facebook server does not support may be specified for <i>param.permissions</i> , or <i>param.fbAppId</i> and the title ID of PARAM.SFO may not be correctly bound.

Value	Hexadecimal	Description
SCE_NP_SNS_FACEBOOK_ERROR_ACCESS_NOT_ALLOWED	0x8055260b	User permission could not be obtained To obtain an access token, in addition to being able to sign into a Facebook account, the user must have access enabled in advance through system software settings.
SCE_NP_SNS_FACEBOOK_ERROR_AUTH_ERROR	0x8055260c	Authentication processing failed with the Facebook account registered to the system.
SCE_NP_SNS_FACEBOOK_ERROR_ACCOUNT_NOT_BOUND	0x8055260d	The Facebook account has not been registered.
SCE_NP_SNS_FACEBOOK_ERROR_CONFIG_DISABLED	0x8055260e	Access from applications is not allowed by system software settings.
SCE_NP_SNS_FACEBOOK_ERROR_CANCELED_BY_SYSTEM	0x8055260f	Communication has been forcibly canceled by the system. This occurs when an application is suspended, etc. Destroy the request.

Description

This function obtains the access token required to access Facebook Graph API.

Using Facebook Graph API provided by Facebook, an application can easily write to, or read user information from Facebook.

To access Facebook Graph API, an access token (user account authentication information) is required. This function enables the obtainment of an access token based on the Facebook account information managed by the system software.

For the application to obtain an access token, it is necessary to obtain a Facebook application ID in advance, and to bind the title ID. The Facebook application ID can be obtained by registering the application on the Facebook developer's site (<https://developers.facebook.com/>). After obtainment, bind the title ID in PARAM.SFO on the Facebook application settings page.

Start up the application with the bound title ID in PARAM.SFO, store the application ID in *param.fbAppId*, and call this function.

Depending on Facebook Graph API, an access token with Extended Permissions will be required. To obtain this, store the string representing the required Extended Permissions (separate multiple Extended Permissions with commas) to *param.permissions*.

Since Facebook Permission Dialog is not displayed with this function, an error code will be returned if the **Allow Access from Applications** setting is not enabled, or if confirmation of the user's Extended Permissions is required.

Note

The URL of reference destinations provided in this document has been checked to be valid as of January 24, 2014. Note that pages may have been moved or their contents modified since then.

Examples

Code Example

```

SceNpSnsFacebookAccessTokenParam param;
SceNpSnsFacebookLongAccessTokenResult result;

// Initialize structures
memset(&param, 0x00, sizeof(param));
memset(&result, 0x00, sizeof(result));
int ret;

// Store the application ID registered to Facebook in param.fbAppId
// Store Extended Permissions to param.permissions as necessary
ret = sceNpSnsFacebookGetLongAccessToken(reqId, &param, &result);
if (ret < 0) {
    // Error handling
}

```

Example Using the Obtained Access Token for Calling a Facebook Graph API

```

std::string url; // URL of the Facebook Graph API

// Facebook Graph API base address
url = "https://graph.facebook.com";

// Path to obtain user information
url += "/me";

// Attach access token to the path
url += "?access_token=";

// Access token obtained with sceNpSnsFacebookGetLongAccessToken() must have
// URI Escape processing performed on it before it is actually used, as follows
url += result.access_token;

// Perform HTTP GET and process the response body of the JSON-encoded string
ret = http_get(url.c_str());
if (ret < 0) {
    // Error handling
}

```

Notes

This function may return an error depending on system status, such as when the user has not registered a Facebook account, etc. Basically, for applications we recommend the method whereby access tokens are obtained by using Facebook Permission Dialog.

When calling `sceNpSnsFacebookAbortRequest()` to abort the processing of this function, it is possible for accessing the basic information of the application to be completed on the Facebook-side.

See Also

`sceNpSnsFacebookCreateRequest()`, `sceNpSnsFacebookDeleteRequest()`,
`sceNpSnsFacebookAbortRequest()`

Facebook Permission Dialog Functions

SCE CONFIDENTIAL

sceNpSnsFacebookDialogParamInit

Initialize boot parameters of Facebook Permission Dialog

Definition

```
#include <np_sns_facebook_dialog.h>
static inline
void sceNpSnsFacebookDialogParamInit(SceNpSnsFacebookDialogParam *param)
{
    sceClibMemset( param, 0x0, sizeof(SceNpSnsFacebookDialogParam) );
    _sceCommonDialogSetMagicNumber( &param->commonParam );
    param->mode = SCE_NP_SNS_FACEBOOK_DIALOG_MODE_LONG_ACCESS_TOKEN;
    param->sdkVersion = SCE_PSP2_SDK_VERSION;
}
```

Arguments

param Boot parameters

Return Values

None

Description

This function initializes the boot parameters specified with `sceNpSnsFacebookDialogInit()`. This function must be called to initialize the boot parameters before `sceNpSnsFacebookDialogInit()` is called.

See Also

`sceNpSnsFacebookDialogInit()`

SCE CONFIDENTIAL

sceNpSnsFacebookDialogInit

Start Facebook Permission Dialog

Definition

```
#include <np_sns_facebook_dialog.h>
SceInt32 sceNpSnsFacebookDialogInit(
    const SceNpSnsFacebookDialogParam *param
);
```

Arguments

param Boot parameters

Return Values

Returns a value of 0 or above indicating the processing status common to Common Dialog for normal termination.

Returns a negative value upon error. The main error codes are indicated below; however, program your application so that it does not malfunction when an error code other than one of those listed below is returned.

Value	Hexadecimal	Description
SCE_NP_SNS_FACEBOOK_DIALOG_ERROR_REQUIRED_MODULES_NOT_LOADED	0x80106102	Necessary module is not loaded Check to see that SCE_SYSMODULE_NP_SNS_FACEBOOK was loaded.
SCE_COMMON_DIALOG_ERROR_INVALID_INFOBAR_PARAM	0x80020433	Value other than NULL was specified to the info bar parameter
SCE_COMMON_DIALOG_ERROR_INVALID_BG_COLOR	0x80020434	Value other than NULL was specified to the background color parameter
SCE_COMMON_DIALOG_ERROR_INVALID_DIMMER_COLOR	0x80020435	Invalid dimmer color was specified

Description

This function starts the processing of Facebook Permission Dialog. This function is a non-blocking function; it requests the start of Facebook Permission Dialog to the system and returns immediately.

After calling this function, call `sceNpSnsFacebookDialogGetStatus()` to obtain the operation status. While communication is being processed or during user operation, the operation status will be `SCE_COMMON_DIALOG_STATUS_RUNNING`. After this, when process succeeds or when an error occurs, the operation status will be updated to `SCE_COMMON_DIALOG_STATUS_FINISHED`. The return value can be obtained with `sceNpSnsFacebookDialogGetResultLongToken()`.

Regardless of the success/failure of the process, call `sceNpSnsFacebookDialogTerm()` to terminate Facebook Permission Dialog once the operation status becomes `SCE_COMMON_DIALOG_STATUS_FINISHED`.

Facebook Permission Dialog will display an appropriate dialog depending on the contents of Facebook-related settings. Also, if the application requires an Extended Permission, Facebook Permission Dialog will request the user to allow Extended Permissions.

SCE CONFIDENTIAL

Notes

In order to use Facebook Permission Dialog, it is necessary to load the `SCE_SYSMODULE_NP_SNS_FACEBOOK` module and to call `sceNpSnsFacebookInit()`.

When Facebook Permission Dialog is called, one of several dialogs will be displayed depending on system status. Refer to the "NP SNS Facebook Library Overview" document concerning the dialogs displayed.

See Also

`sceNpSnsFacebookInit()`, `SCE_COMMON_DIALOG_STATUS_XXX`,
`sceNpSnsFacebookDialogGetStatus()`, `sceNpSnsFacebookDialogGetResultLongToken()`,
`sceNpSnsFacebookDialogTerm()`

SCE CONFIDENTIAL

sceNpSnsFacebookDialogTerm

Terminate Facebook Permission Dialog

Definition

```
#include <np_sns_facebook_dialog.h>
SceInt32 sceNpSnsFacebookDialogTerm(
    void
);
```

Arguments

None

Return Values

Returns a value of 0 or above indicating the processing status common to Common Dialog for normal termination.

Returns a negative value for an error.

Description

This function terminates Facebook Permission Dialog.

Make sure to call this function when SCE_COMMON_DIALOG_STATUS_FINISHED returns.

See Also

sceNpSnsFacebookDialogInit()

SCE CONFIDENTIAL

sceNpSnsFacebookDialogGetStatus

Get the operation status of Facebook Permission Dialog

Definition

```
#include <np_sns_facebook_dialog.h>
SceCommonDialogStatus sceNpSnsFacebookDialogGetStatus (
    void
);
```

Arguments

None

Return Values

Returns a value of 0 or above indicating the processing status common to Common Dialog for normal termination.

Returns a negative value for an error.

Description

This function returns the current operating status of Facebook Permission Dialog.

Value	Decimal	Description
SCE_COMMON_DIALOG_STATUS_NONE	0	Not processed
SCE_COMMON_DIALOG_STATUS_RUNNING	1	Processing
SCE_COMMON_DIALOG_STATUS_FINISHED	2	Processing complete

After calling `sceNpSnsFacebookDialogInit()`, leave a certain interval and then call this function to obtain the operation status.

See Also

`SCE_COMMON_DIALOG_STATUS_XXX`, `sceNpSnsFacebookDialogInit()`

SCE CONFIDENTIAL

sceNpSnsFacebookDialogGetResultLongToken

Get result of Facebook Permission Dialog

Definition

```
#include <np_sns_facebook_dialog.h>
SceInt32 sceNpSnsFacebookDialogGetResultLongToken (
    SceNpSnsFacebookDialogResultLongToken *result
);
```

Arguments

result Pointer to the structure for storing the result

Return Values

Returns a value of 0 or above indicating the processing status common to Common Dialog for normal termination.

Returns a negative value for an error.

Description

After the processing of Facebook Permission Dialog completes, call this function to obtain the result.

If this function's return value is SCE_OK(0), execution results of Facebook Permission Dialog will be stored in the *result* specified in the argument.

The return codes stored in *result* of the *SceNpSnsFacebookDialogResultLongToken* structure are as follows:

Value	Hexadecimal	Description
SCE_COMMON_DIALOG_RESULT_OK	0x0	Normal termination
SCE_COMMON_DIALOG_RESULT_USER_CANCELED	0x1	User canceled the processing
SCE_COMMON_DIALOG_RESULT_ABORTED	0x2	Forced-termination by <i>sceNpSnsFacebookDialogAbort()</i>
SCE_NP_SNS_FACEBOOK_DIALOG_ERROR_NOT_SIGNIN	0x80106104	NP SNS Facebook library functions cannot be used because the user is not signed in.
SCE_NP_SNS_FACEBOOK_DIALOG_ERROR_CHAT_RESTRICTION	0x80106105	NP SNS Facebook library functions cannot be used due to chat restrictions
SCE_NP_SNS_FACEBOOK_DIALOG_ERROR_LIBRARY_NOT_INITIALIZED	0x80106106	NP SNS Facebook library has not been initialized
	Negative value	Error termination Value represents the error code.

See Also

SCE_COMMON_DIALOG_RESULT_XXX, *SceNpSnsFacebookDialogResultLongToken*

SCE CONFIDENTIAL

sceNpSnsFacebookDialogAbort

Abort Facebook Permission Dialog

Definition

```
#include <np_sns_facebook_dialog.h>
SceInt32 sceNpSnsFacebookDialogAbort(
    void
);
```

Arguments

None

Return Values

Returns a value of 0 or above indicating the processing status common to Common Dialog for normal termination.

Returns a negative value for an error.

Description

This function aborts Facebook Permission Dialog.

This function can be called at any time after `sceNpSnsFacebookDialogInit()` and before `sceNpSnsFacebookDialogTerm()`.

When the call succeeds, Facebook Permission Dialog will begin to abort its processing. When the abort completes, the operation status will transition to `SCE_COMMON_DIALOG_STATUS_FINISHED`.

For details on the operation status, refer to the explanation of `sceNpSnsFacebookDialogGetStatus()`.

Even if processing is aborted, it is possible that application access permission or Extended Permissions may appear to have been completed on the Facebook side.

Structures

000004892117

SCE CONFIDENTIAL

SceNpSnsFacebookAccessTokenParam

Parameters to obtain an access token

Definition

```
#include <np/np_sns_facebook.h>
#define SCE_NP_SNS_FACEBOOK_PERMISSIONS_LENGTH_MAX (255)
typedef struct SceNpSnsFacebookAccessTokenParam {
    SceUInt64 fbAppId;
    char permissions[SCE_NP_SNS_FACEBOOK_PERMISSIONS_LENGTH_MAX + 1];
    SceUInt8 reserved[32];
} SceNpSnsFacebookAccessTokenParam;
```

Members

<i>fbAppId</i>	Application ID registered to Facebook
<i>permissions</i>	String of Extended Permissions separated by commas
<i>reserved</i>	Reserved area (fill with all 0s)

Description

This is the structure of the parameters given when Facebook Permission Dialog is started, or when an access token is obtained with `sceNpSnsFacebookGetLongAccessToken()`.

For *fbAppId*, enter the application ID registered in advance to Facebook.

For *permissions*, store the string representing the required Extended Permissions (separate multiple Extended Permissions with commas). Required Extended Permissions differ by Facebook Graph API. For details on Extended Permissions, refer to <http://developers.facebook.com/docs/authentication/permissions>. For each Facebook Graph API specification, such as which Extended Permissions are required, refer to <http://developers.facebook.com/docs/reference/api/>.

See Also

`sceNpSnsFacebookGetLongAccessToken()`, `SceNpSnsFacebookDialogParam`

SceNpSnsFacebookLongAccessTokenResult

Access token and its expiration

Definition

```
#include <np/np_sns_facebook.h>
#define SCE_NP_SNS_FACEBOOK_LONG_ACCESS_TOKEN_LENGTH_MAX (4096)
typedef struct SceNpSnsFacebookLongAccessTokenResult {
    SceUInt64 expiration;
    char accessToken[SCE_NP_SNS_FACEBOOK_LONG_ACCESS_TOKEN_LENGTH_MAX +
1];
    SceUInt8 reserved[39];
} SceNpSnsFacebookLongAccessTokenResult;
```

Members

<i>expiration</i>	Validity period of the access token (seconds)
<i>accessToken</i>	Access token string
<i>reserved</i>	Reserved area

Description

This is a structure for receiving the call result of Facebook Permission Dialog and the access token obtained with `sceNpSnsFacebookGetLongAccessToken()`.

expiration stores the validity period of the access token - in other words, for how many seconds the obtained access token is valid after `sceNpSnsFacebookGetLongAccessToken()` returns. When the validity period expires, a new access token must be obtained.

accessToken stores the string of the obtained access token. Applying the URI Escape processing to this string will enable you to attach Facebook Graph API URL. `sceHttpUriEscape()` provided by libhttp can be used for the URI Escape processing.

Notes

Currently, the validity period is approximately 1 hour to 1 hour and a half. Note, however, that the validity period may be changed from the Facebook-side.

See Also

`sceNpSnsFacebookGetLongAccessToken()`, `sceHttpUriEscape()`

SCE CONFIDENTIAL

SceNpSnsFacebookDialogParam

Boot parameters of Facebook Permission Dialog

Definition

```
#include <np_sns_facebook_dialog.h>
typedef struct SceNpSnsFacebookDialogParam {
    SceUInt32 sdkVersion;
    SceCommonDialogParam commonParam;
    SceNpSnsFacebookAccessTokenParam accessTokenParam;
    SceInt32 mode;
    SceUInt8 reserved[28];
} SceNpSnsFacebookDialogParam;
```

Members

<i>sdkVersion</i>	Area used within the library. The application does not need to directly reference this space.
<i>commonParam</i>	Common parameters for Common Dialogs
<i>accessTokenParam</i>	Parameters for obtaining an access token [IN] Initialize the structure with 0's first, and then set <i>fbAppId</i> , as well as the other required fields.
<i>mode</i>	Area used within the library for compatibility. Do not change this from the value set with <i>sceNpSnsFacebookDialogParamInit()</i> .
<i>reserved</i>	Reserved area (fill with all 0s)

Description

This datatype represents the boot parameters of Facebook Permission Dialog.

It is specified upon initializing the boot parameters with *sceNpSnsFacebookDialogParamInit()* and starting Facebook Permission Dialog with *sceNpSnsFacebookDialogInit()*.

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

Given that it is not possible to perform specifications concerning the info bar and background color in Facebook Permission Dialog, always specify NULL in *commonParam.infobarParam* and *commonParam.bgColor*. If values other than NULL are specified, *sceNpSnsFacebookDialogInit()* will return *SCE_COMMON_DIALOG_ERROR_INVALID_INFOBAR_PARAM* and *SCE_COMMON_DIALOG_ERROR_INVALID_BG_COLOR*, respectively.

See Also

sceNpSnsFacebookDialogParamInit()

SceNpSnsFacebookDialogResultLongToken

Structure for obtaining Facebook Permission Dialog call result

Definition

```
#include <np_sns_facebook_dialog.h>
typedef struct SceNpSnsFacebookDialogResultLongToken {
    SceNpSnsFacebookLongAccessTokenResult accessTokenResult;
    SceInt32 result;
    SceUInt8 reserved[36];
} SceNpSnsFacebookDialogResultLongToken;
```

Members

<i>accessTokenResult</i>	Obtained access token and its expiration information
<i>result</i>	Completion status of Facebook Permission Dialog
<i>reserved</i>	Reserved area (fill with all 0s)

Description

This datatype represents the processing result of Facebook Permission Dialog.
It is specified in `sceNpSnsFacebookDialogGetResultLongToken()`.

See Also

`sceNpSnsFacebookDialogGetResultLongToken()`

Constants

000004892117

SCE CONFIDENTIAL

SCE_NP_SNS_FACEBOOK_MAX_REQUEST_NUM

Maximum number of requests

Definition

```
#include <np.h>
#define SCE_NP_SNS_FACEBOOK_MAX_REQUEST_NUM (4)
```

Description

This constant indicates the maximum number of requests that can exist at one time in the NP SNS Facebook library.

See Also

sceNpSnsFacebookCreateRequest ()

SCE CONFIDENTIAL

SCE_NP_SNS_FACEBOOK_PERMISSIONS_LENGTH_MAX

Maximum length of Extended Permission character strings

Definition

```
#include <np.h>
#define SCE_NP_SNS_FACEBOOK_PERMISSIONS_LENGTH_MAX (255)
```

Description

This constant represents the maximum length of the *permissions* argument (a character string with Extended Permissions separated with comma) of the `SceNpSnsFacebookAccessTokenParam` structure. Byte length does not include the NULL termination character.

See Also

`SceNpSnsFacebookAccessTokenParam`

SCE CONFIDENTIAL

SCE_NP_SNS_FACEBOOK_LONG_ACCESS_TOKEN_LENGTH_MAX

Maximum length of access token character string

Definition

```
#include <np.h>
#define SCE_NP_SNS_FACEBOOK_LONG_ACCESS_TOKEN_LENGTH_MAX (4096)
```

Description

This constant represents the maximum length of the *accessToken* argument of the *SceNpSnsFacebookLongAccessTokenResult* structure. Byte length does not include the NULL termination character.

See Also

SceNpSnsFacebookLongAccessTokenResult

SCE CONFIDENTIAL

SCE_NP_SNS_FACEBOOK_DIALOG_MODE_LONG_ACCESS_TOKEN

Mode for obtaining a 4 KB access token

Definition

```
#include <np_sns_facebook_dialog.h>
#define SCE_NP_SNS_FACEBOOK_DIALOG_MODE_LONG_ACCESS_TOKEN 1
```

Description

This constant is set to *mode* of the Facebook Permission Dialog boot parameters structure `SceNpSnsFacebookDialogParam`. *mode* is an area used within the library for compatibility and a suitable value is set with `sceNpSnsFacebookDialogParamInit()`. Do not set the value of *mode* directly by the application.

See Also

`SceNpSnsFacebookDialogParam`

Return Codes

List of error codes returned by the NP SNS Facebook library

List of error codes returned by the NP SNS Facebook library

Value	Hexadecimal	Description
SCE_NP_SNS_FACEBOOK_ERROR_UNKNOWN	0x80552601	An error not listed below occurred (communication error, for example)
SCE_NP_SNS_FACEBOOK_ERROR_ALREADY_INITIALIZED	0x80552602	Attempted to initialize an already initialized library
SCE_NP_SNS_FACEBOOK_ERROR_NOT_INITIALIZED	0x80552603	Attempted to call API in a state where the library has not yet been initialized
SCE_NP_SNS_FACEBOOK_ERROR_INVALID_ARGUMENT	0x80552604	Argument is invalid
SCE_NP_SNS_FACEBOOK_ERROR_OUT_OF_MEMORY	0x80552605	There is not enough memory
SCE_NP_SNS_FACEBOOK_ERROR_EXCEEDS_MAX	0x80552606	Created more than 4 requests at one time.
SCE_NP_SNS_FACEBOOK_ERROR_NOT_SIGNIN	0x80552607	Not signed in to PSN SM
SCE_NP_SNS_FACEBOOK_ERROR_CHAT_RESTRICTION	0x80552608	Facebook coordination features cannot be used due to chat restrictions
SCE_NP_SNS_FACEBOOK_ERROR_ABORTED	0x80552609	Aborted
SCE_NP_SNS_FACEBOOK_ERROR_FACEBOOK_SERVER_ERROR	0x8055260a	Facebook server returned an error. Title ID of PARAM.SFO and an application ID registered to Facebook may not be correctly bound.
SCE_NP_SNS_FACEBOOK_ERROR_ACCESS_NOT_ALLOWED	0x8055260b	User permission could not be obtained
SCE_NP_SNS_FACEBOOK_ERROR_AUTH_ERROR	0x8055260c	Authentication processing with the Facebook account registered to the system has failed
SCE_NP_SNS_FACEBOOK_ERROR_ACCOUNT_NOT_BOUND	0x8055260d	The Facebook account has not been registered.
SCE_NP_SNS_FACEBOOK_ERROR_CONFIG_DISABLED	0x8055260e	Access from applications is not allowed by system software settings.
SCE_NP_SNS_FACEBOOK_ERROR_CANCELED_BY_SYSTEM	0x8055260f	Communication has been forcibly canceled by the system. This occurs when an application is suspended, etc.

Return code of Facebook Permission Dialog

Common Dialog may return other return codes in addition to the following. For details, refer to the "Common Dialog Reference" document.

Value	Hexadecimal	Description
SCE_NP_SNS_FACEBOOK_DIALOG_ERROR_INTERNAL	0x80106101	Internal error within library occurred (Network Error, etc.)
SCE_NP_SNS_FACEBOOK_DIALOG_ERROR_REQUIRED_MODULES_NOT_LOADED	0x80106102	Necessary module is not loaded
SCE_NP_SNS_FACEBOOK_DIALOG_ERROR_PARAM	0x80106103	Invalid value was specified to a parameter
SCE_NP_SNS_FACEBOOK_DIALOG_ERROR_NOT_SIGNIN	0x80106104	NP SNS Facebook library functions cannot be used because the user is not signed in.
SCE_NP_SNS_FACEBOOK_DIALOG_ERROR_CHAT_RESTRICTION	0x80106105	NP SNS Facebook library functions cannot be used due to chat restrictions.
SCE_NP_SNS_FACEBOOK_DIALOG_ERROR_LIBRARY_NOT_INITIALIZED	0x80106106	NP SNS Facebook library has not been initialized