

IME Dialog Reference

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

Table of Contents

Constants **3**

 Character String Size 4

 Return Codes 5

Datatypes **6**

 ScelmeDialogParam 7

 ScelmeDialogResult 9

Functions **10**

 scelmeDialogParamInit 11

 scelmeDialogInit 12

 scelmeDialogGetStatus 14

 scelmeDialogGetResult 15

 scelmeDialogTerm 16

 scelmeDialogAbort 17

000004892117

Constants

000004892117

Character String Size

Maximum size of character strings of IME Dialog library

Definition

Value	Decimal	Description
SCE_IME_DIALOG_MAX_TITLE_LENGTH	128	Maximum size of character string that can be specified to <i>SceImeDialogParam.title</i>
SCE_IME_DIALOG_MAX_TEXT_LENGTH	2048	Maximum size of character string that can be specified to <i>SceImeDialogParam.maxTextLength</i>

Description

The IME Dialog library has a number of functions that can be displayed through specification of a user specified character string. The maximum size of the character string that can be specified is fixed for each function, so specify the character string within the above values. The value above is the size not including NULL termination.

SCE CONFIDENTIAL

Return Codes

List of return codes returned by IME Dialog library

Definition

Value	Hexadecimal	Description
SCE_COMMON_DIALOG_ERROR_BUSY	0x80020401	Calling another Common Dialog function
SCE_COMMON_DIALOG_ERROR_NULL	0x80020402	NULL was specified as the function's argument
SCE_COMMON_DIALOG_ERROR_NOT_RUNNING	0x80020404	Called during a period other than SCE_COMMON_DIALOG_STAT US_RUNNING
SCE_COMMON_DIALOG_ERROR_NOT_FINISHED	0x80020410	Called during a period other than SCE_COMMON_DIALOG_STAT US_FINISHED
SCE_COMMON_DIALOG_ERROR_NOT_IN_USE	0x80020411	sceImeDialogInit() is not called
SCE_COMMON_DIALOG_ERROR_INVALID_INFOBAR_PARAM	0x80020433	<i>commonParam.infobarParam</i> value is invalid
SCE_COMMON_DIALOG_ERROR_INVALID_BG_COLOR	0x80020434	<i>commonParam.bgColor</i> value is invalid
SCE_COMMON_DIALOG_ERROR_INVALID_DIMMER_COLOR	0x80020435	<i>commonParam.dimmerColor</i> value is invalid
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error
SCE_IME_DIALOG_ERROR_PARAM	0x80102001	Parameter error
SCE_IME_DIALOG_ERROR_INTERNAL	0x80102002	Internal error
SCE_IME_DIALOG_ERROR_INVALID_DIALOG_MODE	0x80102003	<i>dialogMode</i> value is invalid
SCE_IME_DIALOG_ERROR_INVALID_TEXT_BOX_MODE	0x80102004	<i>textBoxMode</i> value is invalid
SCE_IME_DIALOG_ERROR_INVALID_TITLE	0x80102005	<i>title</i> value is invalid
SCE_IME_ERROR_INVALID_TEXT	0x80100706	Content of text is invalid
SCE_IME_ERROR_INVALID_INPUT_METHOD	0x80100712	<i>inputMethod</i> value is invalid
SCE_IME_ERROR_INVALID_SUPPORTED_LANGUAGES	0x80100713	<i>supportedLanguages</i> value is invalid
SCE_IME_ERROR_INVALID_TYPE	0x80100714	<i>type</i> value is invalid
SCE_IME_ERROR_INVALID_OPTION	0x80100715	<i>option</i> value is invalid
SCE_IME_ERROR_INVALID_MAX_TEXT_LENGTH	0x80100719	<i>maxTextLength</i> value is invalid
SCE_IME_ERROR_INVALID_INPUT_TEXT_BUFFER	0x8010071a	<i>inputTextBuffer</i> value is invalid
SCE_IME_ERROR_INVALID_RESERVED	0x8010071b	<i>reserved</i> value is invalid
SCE_IME_ERROR_INVALID_ENTER_LABEL	0x8010071c	<i>enterLabel</i> value is invalid

Datatypes

000004892117

SCE CONFIDENTIAL

ScelmeDialogParam

Initialization parameter

Definition

```
#include <ime_dialog.h>
typedef struct SceImeDialogParam {
    SceUInt32 sdkVersion;
    SceUInt32 inputMethod;
    SceUInt64 supportedLanguages;
    SceBool languagesForced;
    SceUInt32 type;
    SceUInt32 option;
    SceImeTextFilter filter;
    SceUInt32 dialogMode;
    SceUInt32 textBoxMode;
    const SceWChar16 *title;
    SceUInt32 maxTextLength;
    SceWChar16 *initialText;
    SceWChar16 *inputTextBuffer;
    SceCommonDialogParam commonParam;
    SceUChar8 enterLabel;
    SceChar8 reserved[35];
} SceImeDialogParam;
```

Members

<i>sdkVersion</i>	SDK version
<i>inputMethod</i>	Input method
<i>supportedLanguages</i>	Supported languages
<i>languagesForced</i>	Whether to forcibly set supported languages
<i>type</i>	Input type
<i>option</i>	Input options
<i>filter</i>	Character strings filter
<i>dialogMode</i>	Dialog mode
<i>textBoxMode</i>	Textbox mode
<i>title</i>	Title
<i>maxTextLength</i>	Maximum length of input character string
<i>initialText</i>	Initial text of the input character string
<i>inputTextBuffer</i>	Pointer to the storage location of input result character string
<i>commonParam</i>	Common parameters for Common Dialogs
<i>enterLabel</i>	Enter button label
<i>reserved</i>	Reserved area (fill with all 0s)

Description

This structure is used to specify parameters when initializing the library with `sceImeDialogInit()`.

Refer to the following, concerning *inputMethod*, *supportedLanguages*, *languagesForced*, *type*, *option*, *filter* and *enterLabel*.

- The "SceImeParam" section in the "libime Reference" document

Set the setting for the IME Dialog cancel operation in *dialogMode*. The following values can be input.

Value	Decimal	Description
SCE_IME_DIALOG_DIALOG_MODE_DEFAULT	0	With no cancel operation
SCE_IME_DIALOG_DIALOG_MODE_WITH_CANCEL	1	With cancel operation

Set the textbox setting to be used in IME Dialog in *textBoxMode*. The following values can be input.

Value	Decimal	Description
SCE_IME_DIALOG_TEXTBOX_MODE_DEFAULT	0	Textbox for regular sentence input
SCE_IME_DIALOG_TEXTBOX_MODE_PASSWORD	1	Textbox for password input
SCE_IME_DIALOG_TEXTBOX_MODE_WITH_CLEAR	2	Textbox with clear button

In *title*, set the character string displayed in the title part of IME Dialog. Substitute 0 for the termination of the character string.

Set maximum length of the input character string in *maxTextLength*.

In *initialText*, set the pointer to the character string set from the beginning when the dialog is displayed. Substitute 0 for the termination of the character string.

Set the pointer to the storage location of the input result character string in *inputTextBuffer*. Since, with `SceWChar16` type, 0 is added at the end of the character string, area length must be *maxTextLength* + 1.

commonParam stores parameters that are common to Common Dialogs. For details on the common parameters, refer to the "Common Dialog Reference" document.

Fill *reserved* with all 0s.

See Also

`sceImeDialogInit()`

SCE CONFIDENTIAL

ScelmeDialogResult

IME Dialog call result

Definition

```
#include <ime_dialog.h>
typedef struct SceImeDialogResult {
    SceInt32 result;
    SceInt32 button;
    SceChar8 reserved[28];
} SceImeDialogResult;
```

Members

result Result
button Selected button
reserved Reserved area (fill with all 0s)

Description

This structure stores IME Dialog operation results. Specify as the argument of `sceImeDialogGetResult()`.

One of the following values is input in *result*.

Value	Hexadecimal	Description
SCE_COMMON_DIALOG_RESULT_OK	0x0	User selected either close button or Enter button
SCE_COMMON_DIALOG_RESULT_USER_CANCELED	0x1	User performed cancel operation.
SCE_COMMON_DIALOG_RESULT_ABORTED	0x2	IME Dialog operation has been aborted through <code>sceImeDialogAbort()</code>

One of the following values is input in *button*.

Value	Hexadecimal	Description
SCE_IME_DIALOG_BUTTON_NONE	0x0	IME Dialog operation has been aborted through cancel operation or <code>sceImeDialogAbort()</code>
SCE_IME_DIALOG_BUTTON_CLOSE	0x1	User selected close button
SCE_IME_DIALOG_BUTTON_ENTER	0x2	User selected Enter button

See Also

`sceImeDialogGetResult()`

Functions

000004892117

SCE CONFIDENTIAL

sceImeDialogParamInit

Initialize parameter

Definition

```
#include <ime_dialog.h>
void sceImeDialogParamInit(
    SceImeDialogParam *param
)
```

Arguments

param Initialization parameters

Return Values

None

Description

This is a function for initializing the IME Dialog call parameters.

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

Examples

```
SceImeDialogParam param;

sceImeDialogParamInit( &param );
param.maxTextLength = MAX_TEXT_LENGTH;
param.initialText = initialText;
param.inputTextBuffer = inputTextBuffer;
if ( sceImeDialogInit( &param ) < 0 ) {
    // error
}
```

See Also

SceImeDialogParam, sceImeDialogInit()

SCE CONFIDENTIAL

sceImeDialogInit

Initialize IME Dialog

Definition

```
#include <ime_dialog.h>
SceInt32 sceImeDialogInit(
    const SceImeDialogParam *param
)
```

Arguments

param Initialization parameters

Return Values

Returns SCE_OK(0) for success.

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

Value	Hexadecimal	Description
SCE_COMMON_DIALOG_ERROR_BUSY	0x80020401	The Common Dialog function is already being called (details below)
SCE_COMMON_DIALOG_ERROR_NULL	0x80020402	NULL was specified for the argument <i>param</i>
SCE_COMMON_DIALOG_ERROR_INVALID_INFOBAR_PARAM	0x80020433	<i>commonParam.infobarParam</i> value is invalid
SCE_COMMON_DIALOG_ERROR_INVALID_BG_COLOR	0x80020434	<i>commonParam.bgColor</i> value is invalid
SCE_COMMON_DIALOG_ERROR_INVALID_DIMMER_COLOR	0x80020435	<i>commonParam.dimmerColor</i> value is invalid
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error
SCE_IME_DIALOG_ERROR_PARAM	0x80102001	Parameter error
SCE_IME_DIALOG_ERROR_INTERNAL	0x80102002	Internal error
SCE_IME_DIALOG_ERROR_INVALID_DIALOG_MODE	0x80102003	<i>dialogMode</i> value is invalid
SCE_IME_DIALOG_ERROR_INVALID_TEXT_BOX_MODE	0x80102004	<i>textBoxMode</i> value is invalid
SCE_IME_DIALOG_ERROR_INVALID_TITLE	0x80102005	<i>title</i> value is invalid
SCE_IME_ERROR_INVALID_TEXT	0x80100706	Content of text is invalid
SCE_IME_ERROR_INVALID_INPUT_METHOD	0x80100712	<i>inputMethod</i> value is invalid
SCE_IME_ERROR_INVALID_SUPPORTED_LANGUAGES	0x80100713	<i>supportedLanguages</i> value is invalid
SCE_IME_ERROR_INVALID_TYPE	0x80100714	<i>type</i> value is invalid
SCE_IME_ERROR_INVALID_OPTION	0x80100715	<i>option</i> value is invalid
SCE_IME_ERROR_INVALID_MAX_TEXT_LENGTH	0x80100719	<i>maxTextLength</i> value is invalid
SCE_IME_ERROR_INVALID_INPUT_TEXT_BUFFER	0x8010071a	<i>inputTextBuffer</i> value is invalid
SCE_IME_ERROR_INVALID_RESERVED	0x8010071b	<i>reserved</i> value is invalid

Description

This is a function for initializing IME Dialog

It can only be called when other Common Dialog functions are not being called (including functions of IME Dialog itself). If called at other times, `SCE_COMMON_DIALOG_ERROR_BUSY` is returned.

If calling is successful, operation status will immediately change to

`SCE_COMMON_DIALOG_STATUS_RUNNING`. Refer to the `sceImeDialogGetStatus()` section for details on operation status.

In *param*, specify the initialization parameter structure where operation mode, initial values, buffer for storing character strings etc. are set.

Always initialize *param* with the `sceImeDialogParamInit()` macro before setting the various values.

It is not necessary to allocate *param*'s instance after calling this function, but some pointer reference parameters must be saved until calling the IME Dialog function has been completed with `sceImeDialogTerm()`.

This function is multithread safe.

Examples

```
SceImeDialogParam param;

sceImeDialogParamInit( &param );
param.maxTextLength = MAX_TEXT_LENGTH;
param.initialText = initialText;
param.inputTextBuffer = inputTextBuffer;
if ( sceImeDialogInit( &param ) < 0 ) {
    // error
}
```

See Also

`SceImeDialogParam`, `sceImeDialogParamInit()`

sceImeDialogGetStatus

Get operation status of IME Dialog

Definition

```
#include <ime_dialog.h>
SceCommonDialogStatus sceImeDialogGetStatus()
```

Arguments

None

Return Values

Returns one of the following operation statuses.

Value	Hexadecimal	Description
SCE_COMMON_DIALOG_STATUS_NONE	0x0	IME Dialog is not running
SCE_COMMON_DIALOG_STATUS_RUNNING	0x1	IME Dialog is running
SCE_COMMON_DIALOG_STATUS_FINISHED	0x2	IME Dialog operation has finished

Description

This function gets the operation status of IME Dialog.

The default value of the operation status is SCE_COMMON_DIALOG_STATUS_NONE.

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

`sceImeDialogTerm()` can be called only while the operation status is SCE_COMMON_DIALOG_STATUS_RUNNING.

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

This function is multithread safe.

See Also

`sceImeDialogInit()`, `sceImeDialogTerm()`

SCE CONFIDENTIAL

sceImeDialogGetResult

Get call result of IME Dialog

Definition

```
#include <ime_dialog.h>
SceInt32 sceImeDialogGetResult(
    SceImeDialogResult *result
)
```

Arguments

result Structure that stores the result

Return Values

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

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

Value	Hexadecimal	Description
SCE_COMMON_DIALOG_ERROR_NULL	0x80020402	NULL was specified in the argument <i>result</i>
SCE_COMMON_DIALOG_ERROR_NOT_FINISHED	0x80020410	Called during other than the appropriate operation status (details below)
SCE_IME_ERROR_INVALID_RESERVED	0x8010071b	<i>reserved</i> value is invalid (details below)
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

Description

This function retrieves the call result of IME Dialog.

This function can be called only while the operation status of IME 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.

If `SceImeDialogResult.reserved` is not filled with 0s, SCE_IME_ERROR_INVALID_RESERVED is returned as the value of the function.

This function is multithread safe.

See Also

`sceImeDialogInit()`, `sceImeDialogTerm()`

SCE CONFIDENTIAL

sceImeDialogTerm

Terminate IME Dialog

Definition

```
#include <ime_dialog.h>
SceInt32 sceImeDialogTerm()
```

Arguments

None

Return Values

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

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

Value	Hexadecimal	Description
SCE_COMMON_DIALOG_ERROR_NOT_FINISHED	0x80020410	Called during other than the appropriate operation status (details below)
SCE_COMMON_DIALOG_ERROR_NOT_IN_USE	0x80020411	sceImeDialogInit() is not called
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

Description

This function ends calling of IME Dialog. Calling must be ended with this function after IME Dialog has been called with sceImeDialogInit().

This function can be called only while the operation status of IME 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. SCE_COMMON_DIALOG_ERROR_NOT_IN_USE will be returned if the IME Dialog function 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 sceImeDialogGetStatus() section.

This function is multithread safe.

See Also

sceImeDialogInit(), sceImeDialogGetStatus()

SCE CONFIDENTIAL

sceImeDialogAbort

Abort call of IME Dialog

Definition

```
#include <ime_dialog.h>
SceInt32 sceImeDialogAbort()
```

Arguments

None

Return Values

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

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

Value	Hexadecimal	Description
SCE_COMMON_DIALOG_ERROR_NOT_IN_USE	0x80020411	sceImeDialogInit() is not called
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

Description

This function aborts calling of IME Dialog.

It can be called at any time between when sceImeDialogInit() is called and sceImeDialogTerm() 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, IME Dialog will begin to abort processing.

When processing is completely aborted, the operation status will change to SCE_COMMON_DIALOG_STATUS_FINISHED.

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

When IME Dialog is closed with this function, the *result* of SceImeDialogResult will be SCE_COMMON_DIALOG_RESULT_ABORTED with calling sceImeDialogGetResult().

sceImeDialogAbort() is used to promptly abort the IME Dialog display, for example when an urgent interrupt must be processed.

This function is multithread safe.

See Also

sceImeDialogInit(), sceImeDialogTerm()