

Save Data Dialog Reference

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

Table of Contents

Save Data Dialog Call	4
sceSaveDataDialogParamInit	5
sceSaveDataDialogInit	6
SceSaveDataDialogParam	8
SceSaveDataDialogFixedParam	11
SceSaveDataDialogListParam	12
SceSaveDataDialogUserMessageParam	14
SceSaveDataDialogSystemMessageParam	15
SceSaveDataDialogErrorCodeParam	17
SceSaveDataDialogProgressBarParam	18
SceSaveDataDialogSlotConfigParam	20
SceSaveDataDialogAppSubDirParam	21
Obtaining Operation Status	22
sceSaveDataDialogGetStatus	23
sceSaveDataDialogGetSubStatus	24
Updating Display Information	25
sceSaveDataDialogProgressBarInc	26
sceSaveDataDialogProgressBarSetValue	28
Closing Save Data Dialog	30
sceSaveDataDialogSubClose	31
Retrieval of Save Data Dialog Call Result	33
sceSaveDataDialogGetResult	34
SceSaveDataDialogResult	36
SceSaveDataDialogSlotInfo	38
Continuous Call of Save Data Dialog	39
sceSaveDataDialogContinue	40
sceSaveDataDialogFinish	42
SceSaveDataDialogFinishParam	44
Aborting Save Data Dialog	45
sceSaveDataDialogAbort	46
Terminating Save Data Dialog	48
sceSaveDataDialogTerm	49
Save Data Dialog Call for Save Data for PSP™	51
scePspSaveDataDialogParamInit	52
scePspSaveDataDialogInit	53
ScePspSaveDataDialogParam	55
ScePspSaveDataDialogFixedParam	57
ScePspSaveDataDialogListParam	58
ScePspSaveDataDialogUserMessageParam	60
ScePspSaveDataDialogSystemMessageParam	61
ScePspSaveDataDialogErrorCodeParam	62
ScePspSaveDataDialogProgressBarParam	63
Obtaining Save Data Dialog Call Result for Save Data for PSP™	64

SCE CONFIDENTIAL

scePspSaveDataDialogGetResult	65
ScePspSaveDataDialogResult	66
SceAppUtilPspSaveDataParamSfo	67
Continuous Call of Save Data Dialog for Save Data for PSP™	68
scePspSaveDataDialogContinue	69
Terminating Save Data Dialog for Save Data for PSP™	71
scePspSaveDataDialogTerm	72
Constants	73
Character string size	74
Save Data Slot Maximum Size	75
Return Codes	76
Parameter errors	77
Specification of Save Data	80
SceAppUtilSaveDataSlotId	81
SceAppUtilSaveDataSlotEmptyParam	82
SceAppUtilSaveDataSlot	83
SceAppUtilMountPoint	84
Specification of Save Data for PSP™	85
SceAppUtilPspSaveDataDirName	86

Save Data Dialog Call

SCE CONFIDENTIAL

sceSaveDataDialogParamInit

Macro for call parameter initialization

Definition

```
#include <savedata_dialog.h>
static inline
void sceSaveDataDialogParamInit(SceSaveDataDialogParam *param)
{
    sceClibMemset( param, 0x0, sizeof(SceSaveDataDialogParam) );
    _sceCommonDialogSetMagicNumber( &param->commonParam );
    param->sdkVersion = SCE_PSP2_SDK_VERSION;
}
```

Arguments

param Call parameter

Return Values

None

Description

This is a macro function for initializing the Save Data 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 at the same time.

Examples

```
SceSaveDataDialogParam param;
SceSaveDataDialogSystemMessageParam sysMsgParam;

sceSaveDataDialogParamInit( &param );
sceClibMemset( &sysMsgParam, 0, sizeof(sysMsgParam) );
param.mode = SCE_SAVEDATA_DIALOG_MODE_SYSTEM_MSG;
param.sysMsgParam = &sysMsgParam;
param.sysMsgParam->sysMsgType = SCE_SAVEDATA_DIALOG_SYSMMSG_TYPE_CONFIRM;

if ( sceSaveDataDialogInit( &param ) < 0 ) {
    // error
}
```

See Also

SceSaveDataDialogParam

SCE CONFIDENTIAL

sceSaveDataDialogInit

Call various functions of Save Data Dialog

Definition

```
#include <savedata_dialog.h>
SceInt32 sceSaveDataDialogInit(
    const SceSaveDataDialogParam *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 function is already being called (details below)
SCE_COMMON_DIALOG_ERROR_NULL	0x80020402	NULL was specified for the <i>param</i> argument
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

Description

This function calls the Save Data Dialog function.

This function can be called only when other Common Dialog functions are not called (including the function of Save Data Dialog proper). If this function is called at times other than the above, SCE_COMMON_DIALOG_ERROR_BUSY is returned. When calling this function is successful, the operation status immediately changes to SCE_COMMON_DIALOG_STATUS_RUNNING. For details on the operation statuses, refer to the sceSaveDataDialogGetStatus() section.

In *param*, specify the call parameter structure for which the operation mode, the display target save data ID, display character string, etc. were set.

Be sure to set values for *param* after performing initialization with the sceSaveDataDialogParamInit() macro.

The *param* instance need not be allocated after this function is called, but some of the pointer reference parameters must be held until calling of the Save Data Dialog function is ended by calling sceSaveDataDialogTerm(). For parameters which must be held, refer to the descriptions for individual parameters. Parameters can be released after the function is called unless otherwise noted.

If the content of *param* is invalid, a parameter error will occur, and SCE_SAVEDATA_DIALOG_ERROR_PARAM will return to SceSaveDataDialogResult.result, which is obtained with sceSaveDataDialogGetResult(). As an exception, only when *param.commonParam.bgColor* (background color) is not set to NULL, the API immediately returns SCE_SAVEDATA_DIALOG_ERROR_PARAM, resulting in a failure of the call.

This function is multithread safe.

SCE CONFIDENTIAL

Examples

```
SceSaveDataDialogParam param;
SceSaveDataDialogUserMessageParam userMsgParam;
static const SceChar8 msg_hello[] = "Hello world!";

sceSaveDataDialogParamInit( &param );
sceClibMemset( &userMsgParam, 0, sizeof(userMsgParam) );
param.mode = SCE_SAVEDATA_DIALOG_MODE_USER_MSG;
param.userMsgParam = &userMsgParam;
param.userMsgParam->msg = msg_hello;
param.userMsgParam->buttonType = SCE_SAVEDATA_DIALOG_BUTTON_TYPE_OK;

if ( sceSaveDataDialogInit( &param ) < 0 ) {
    //error
}
```

See Also

SceSaveDataDialogParam, sceSaveDataDialogParamInit(),
sceSaveDataDialogGetStatus(), sceSaveDataDialogGetResult()

SCE CONFIDENTIAL

SceSaveDataDialogParam

Structure for calling Save Data Dialog

Definition

```
#include <savedata_dialog.h>
typedef struct SceSaveDataDialogParam {
    SceUInt32 sdkVersion;
    SceCommonDialogParam commonParam;
    SceSaveDataDialogMode mode;
    SceSaveDataDialogType dispType;
    SceSaveDataDialogFixedParam *fixedParam;
    SceSaveDataDialogListParam *listParam;
    SceSaveDataDialogUserMessageParam *userMsgParam;
    SceSaveDataDialogSystemMessageParam *sysMsgParam;
    SceSaveDataDialogErrorCodeParam *errorCodeParam;
    SceSaveDataDialogProgressBarParam *progBarParam;
    SceSaveDataDialogSlotConfigParam *slotConfParam;
    SceSaveDataDialogEnvFlag flag;
    void *userdata;
    SceChar8 reserved[32];
} SceSaveDataDialogParam;
```

Members

<i>sdkVersion</i>	SDK version
<i>commonParam</i>	Common parameters
<i>mode</i>	Operation mode (details below)
<i>dispType</i>	Display wording type (details below)
<i>fixedParam</i>	Save data fixed display parameter
<i>listParam</i>	Save data list display parameter
<i>userMsgParam</i>	User specified message display parameter
<i>sysMsgParam</i>	System defined message display parameter
<i>errorCodeParam</i>	Error code display parameter
<i>progBarParam</i>	Progress bar display parameter
<i>slotConfParam</i>	Slot setting parameter
<i>flag</i>	Environmental setting flag (details below)
<i>userdata</i>	User specified data
<i>reserved</i>	Reserved area (fill with all 0s)

Description

This is a structure passed to `sceSaveDataDialogInit()` to display Save Data Dialog. Use `sceSaveDataDialogParamInit()` to initialize it.

Specify the SDK version in *sdkVersion*. An appropriate value is input when the structure is initialized with `sceSaveDataDialogParamInit()`.

The display state of info bar and dimmer color can be specified with *commonParam*; however, the background color cannot be specified. For details, refer to the "Common Dialog Reference" document.

Specify the operation mode in *mode*. One of the following values is input.

Value	(Number)	Description
SCE_SAVEDATA_DIALOG_MODE_FIXED	1	Save data fixed display
SCE_SAVEDATA_DIALOG_MODE_LIST	2	Save data list display
SCE_SAVEDATA_DIALOG_MODE_USER_MSG	3	User specified message display
SCE_SAVEDATA_DIALOG_MODE_SYSTEM_MSG	4	System defined message display
SCE_SAVEDATA_DIALOG_MODE_ERROR_CODE	5	Error code display
SCE_SAVEDATA_DIALOG_MODE_PROGRESS_BAR	6	Progress bar display

Specify the display wording type of Save Data Dialog in *dispType*. One of the following values is input.

Value	(Number)	Description
SCE_SAVEDATA_DIALOG_TYPE_SAVE	1	Save wording display
SCE_SAVEDATA_DIALOG_TYPE_LOAD	2	Load wording display
SCE_SAVEDATA_DIALOG_TYPE_DELETE	3	Delete wording display

For example, if SCE_SAVEDATA_DIALOG_SYMSG_TYPE_CONFIRM is specified with the system defined message display setting described later, one of the following messages will be displayed:

- In the case of SCE_SAVEDATA_DIALOG_TYPE_SAVE, "Do you want to save the data?"
- In the case of SCE_SAVEDATA_DIALOG_TYPE_LOAD, "Do you want to load this saved data?"
- In the case of SCE_SAVEDATA_DIALOG_TYPE_DELETE, "Do you want to delete this saved data?"

The structure storing the save data fixed display setting is passed to *fixedParam*. For details, refer to the *SceSaveDataDialogFixedParam* section. If an operation mode other than SCE_SAVEDATA_DIALOG_MODE_FIXED was specified, this value must be set to NULL.

The structure storing the save data list display setting is passed to *listParam*. For details, refer to the *SceSaveDataDialogListParam* section. If an operation mode other than SCE_SAVEDATA_DIALOG_MODE_LIST was specified, this value must be set to NULL.

The structure storing the user specified message display setting is passed to *userMsgParam*. For details, refer to the *SceSaveDataDialogUserMessageParam* section.

If an operation mode other than SCE_SAVEDATA_DIALOG_MODE_USER_MSG was specified, this value must be set to NULL.

The structure storing the system defined message display setting is passed to *sysMsgParam*. For details, refer to the *SceSaveDataDialogSystemMessageParam* section.

If an operation mode other than SCE_SAVEDATA_DIALOG_MODE_SYSTEM_MSG was specified, this value must be set to NULL.

The structure storing the error code display setting is passed to *errorCodeParam*. For details, refer to the *SceSaveDataDialogErrorCodeParam* section. If an operation mode other than SCE_SAVEDATA_DIALOG_MODE_ERROR_CODE was specified, this value must be set to NULL.

The structure storing the progress bar display setting is passed to *progBarParam*.

For details, refer to the *SceSaveDataDialogProgressBarParam* section.

If an operation mode other than SCE_SAVEDATA_DIALOG_MODE_PROGRESS_BAR was specified, this value must be set to NULL.

The structure containing slot settings is passed to *slotConfParam*. If specification is not necessary, set to NULL.

For details, refer to the *SceSaveDataDialogSlotConfigParam* section.

The operating environment of Save Data Dialog is specified in *flag*. The following options can be specified in this SDK.

Value	(Number)	Description
SCE_SAVEDATA_DIALOG_ENV_FLAG_DEFAULT	0	Default setting

The value specified in *userdata* can be received as `SceSaveDataDialogResult.userdata` when `sceSaveDataDialogGetResult()` is called.

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

See Also

`sceSaveDataDialogParamInit()`, `sceSaveDataDialogGetResult()`,
`SceSaveDataDialogFixedParam`, `SceSaveDataDialogUserMessageParam`,
`SceSaveDataDialogSystemMessageParam`, `SceSaveDataDialogErrorCodeParam`,
`SceSaveDataDialogProgressBarParam`, `SceSaveDataDialogSlotConfigParam`,
`SceSaveDataDialogResult`

SCE CONFIDENTIAL

SceSaveDataDialogFixedParam

Structure for fixed save data display

Definition

```
#include <savedata_dialog.h>
typedef struct SceSaveDataDialogFixedParam {
    SceAppUtilSaveDataSlot targetSlot;
    SceChar8 reserved[32];
} SceSaveDataDialogFixedParam;
```

Members

targetSlot Specifies save data slot ID of display target save data (details below)
reserved Reserved area (fill with all 0s)

Description

This structure is used to perform the setting for the fixed save data display. It is used when calling Save Data Dialog in the SCE_SAVEDATA_DIALOG_MODE_FIXED mode. Fill it with all 0s during initialization.

The save data specified in *targetSlot* is displayed on the screen along with thumbnails and other information. For details on the specification of the save data to be displayed, refer to the "Specification of Save Data" chapter.

The definition of the *SceAppUtilSaveDataSlot* structure is located in `include/apputil.h`. For details, refer to the "Application Utility Reference" document.

The save data specified here is displayed on the screen along with thumbnails and other information.

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

See Also

`sceSaveDataDialogInit()`, `SceAppUtilSaveDataSlot`

SceSaveDataDialogListParam

Structure for save data list display

Definition

```
#include <savedata_dialog.h>
typedef struct SceSaveDataDialogListParam {
    const SceAppUtilSaveDataSlot *slotList;
    SceUInt slotListSize;
    SceSaveDataDialogFocusPos focusPos;
    SceAppUtilSaveDataSlotId focusId;
    const SceChar8 *listTitle;
    SceSaveDataDialogListItemStyle itemStyle;
    SceChar8 reserved[28];
} SceSaveDataDialogListParam;
```

Members

<i>slotList</i>	Specifies the IDs of the list display target save data with a list (details below)
<i>slotListSize</i>	Specifies the number of items of the list of save data specified in <i>dataList</i>
<i>focusPos</i>	Specifies the initial focus position on the list (details below)
<i>focusId</i>	Specifies the initial focus position during save data ID specification
<i>listTitle</i>	Specifies the title of the list (details below)
<i>itemStyle</i>	Specifies the display style of the listed item (details below)
<i>reserved</i>	Reserved area (fill with all 0s)

Description

This structure is used to perform the settings for save data list display. It is used when calling Save Data Dialog in the SCE_SAVEDATA_DIALOG_MODE_LIST mode. Fill it with all 0s during initialization.

The save data specified in *slotList* is displayed on the screen along with thumbnails and other information. For details on the specification of the save data to be displayed, refer to the "Specification of Save Data" chapter. The *slotList*'s referent must be held until *sceSaveDataDialogTerm()* has been called, and the Save Data Dialog function call has been completed.

The number of units of save data specified in *slotListSize* is searched and displayed from the head of *slotList*. The maximum number that can be specified is defined by SCE_SAVEDATA_DIALOG_SLOTLIST_MAXSIZE.

The following value is input in *focusPos*.

Value	(Number)	Description
SCE_SAVEDATA_DIALOG_FOCUS_POS_LISTHEAD	0	List head
SCE_SAVEDATA_DIALOG_FOCUS_POS_LISTTAIL	1	List tail
SCE_SAVEDATA_DIALOG_FOCUS_POS_EMPTYHEAD	2	Empty slot head
SCE_SAVEDATA_DIALOG_FOCUS_POS_EMPTYTAIL	3	Empty slot tail
SCE_SAVEDATA_DIALOG_FOCUS_POS_DATAHEAD	4	Data head
SCE_SAVEDATA_DIALOG_FOCUS_POS_DATATAIL	5	Data tail
SCE_SAVEDATA_DIALOG_FOCUS_POS_DATAATEST	6	Latest data
SCE_SAVEDATA_DIALOG_FOCUS_POS_DATAOLDEST	7	Oldest data
SCE_SAVEDATA_DIALOG_FOCUS_POS_ID	8	Slot ID specification

For *focusId*, specify the focus target save data when SCE_SAVEDATA_DIALOG_FOCUS_POS_ID is specified to *focusPos*. If the specified save data is not saved in the list, the focus corresponds to the list head.

The title of the save data list to be displayed can be specified with UTF-8 character string in *listTitle*. String termination must be NULL. Line breaks cannot be inserted. The maximum size is SCE_SAVEDATA_DIALOG_LIST_TITLE_SIZE. If NULL is specified, a system default message will be displayed in accordance with the specified value of *SceSaveDataDialogParam.dispType*.

The display style of the listed item can be specified in *itemStyle*. The following values are input.

Value	(Number)	Description
SCE_SAVEDATA_DIALOG_LIST_ITEM_STYLE_TITLE_DATE_SUBTITLE	0	Performs display on three lines, in the order of Title/Date/Subtitle.
SCE_SAVEDATA_DIALOG_LIST_ITEM_STYLE_TITLE_SUBTITLE_DATE	1	Performs display on three lines, in the order of Title/Subtitle/Date.
SCE_SAVEDATA_DIALOG_LIST_ITEM_STYLE_TITLE_DATE	2	Performs display on two lines, in the order of Title/Date.

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

See Also

`sceSaveDataDialogInit()`, `SceAppUtilSaveDataSlot`,
`SCE_SAVEDATA_DIALOG_LIST_TITLE_SIZE`, `SCE_SAVEDATA_DIALOG_SLOTLIST_MAXSIZE`

SCE CONFIDENTIAL

SceSaveDataDialogUserMessageParam

Structure for user specified message display

Definition

```
#include <savedata_dialog.h>
typedef struct SceSaveDataDialogUserMessageParam {
    SceSaveDataDialogButtonType buttonType;
    const SceChar8 *msg;
    SceAppUtilSaveDataSlot targetSlot;
    SceChar8 reserved[32];
} SceSaveDataDialogUserMessageParam;
```

Members

buttonType Specifies the type of button displayed on the dialog (details below)

msg Arbitrary character string displayed on the screen (NULL termination, UTF-8)

targetSlot Specifies the save data slot ID of the display target save data (details below)

reserved Reserved area (fill with all 0s)

Description

This structure is used to perform the setting for the user specified message display.

It is used when calling Save Data Dialog in the SCE_SAVEDATA_DIALOG_MODE_USER_MSG mode. Fill it with all 0s during initialization.

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

Value	(Number)	Description
SCE_SAVEDATA_DIALOG_BUTTON_TYPE_OK	0	Displays 1 button, the OK button
SCE_SAVEDATA_DIALOG_BUTTON_TYPE_YESNO	1	Displays 2 buttons, the Yes button and the No button
SCE_SAVEDATA_DIALOG_BUTTON_TYPE_NONE	2	Does not display buttons

Save Data Dialog for which SCE_SAVEDATA_DIALOG_BUTTON_TYPE_NONE has been specified cannot be closed by user operation. It must be closed from the caller side by using `sceSaveDataDialogSubClose()` at an arbitrary timing.

In *msg*, specify the character string to be displayed on the screen with UTF-8. The character string termination must be NULL. Up to three line breaks can be inserted and the maximum size is SCE_SAVEDATA_DIALOG_USER_MSG_SIZE. When a string exceeding the maximum size is specified, the exceeding section will be cut off.

The save data specified in *targetSlot* is displayed on the screen along with thumbnails and other information. For details on the specification of the save data to be displayed, refer to the "Specification of Save Data" chapter.

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

See Also

`sceSaveDataDialogInit()`, `sceSaveDataDialogSubClose()`
`SceSaveDataDialogParam`, SCE_SAVEDATA_DIALOG_USER_MSG_SIZE

SCE CONFIDENTIAL

SceSaveDataDialogSystemMessageParam

Structure for system defined message display

Definition

```
#include <savedata_dialog.h>
typedef struct SceSaveDataDialogSystemMessageParam {
    SceSaveDataDialogSystemMessageType sysMsgType;
    SceInt32 value;
    SceAppUtilSaveDataSlot targetSlot;
    SceChar8 reserved[32];
} SceSaveDataDialogSystemMessageParam;
```

Members

sysMsgType Specifies the type of message to be displayed (details below)
value Specifies a value during a message call requiring an additional value
targetSlot Specifies the save data slot ID of the display target save data (details below)
reserved Reserved area (fill with all 0s)

Description

This structure is used to perform the setting for the system defined message display. It is used when calling Save Data Dialog in the `SCE_SAVEDATA_DIALOG_MODE_SYSTEM_MSG` mode. Fill it with all 0s during initialization.

One of the following values is set in *sysMsgType*. The message to be displayed will automatically switch between saving, loading, etc. based on the value set for `SceSaveDataDialogParam.dispType`.

Value	(Number)	Description
<code>SCE_SAVEDATA_DIALOG_SYSMMSG_TYPE_NODATA</code>	1	Displays "There is no saved data."
<code>SCE_SAVEDATA_DIALOG_SYSMMSG_TYPE_CONFIRM</code>	2	Displays "Do you want to save the data?", "Do you want to load this saved data?" or "Do you want to delete this saved data?"
<code>SCE_SAVEDATA_DIALOG_SYSMMSG_TYPE_OVERWRITE</code>	3	Displays "Do you want to overwrite this saved data?"
<code>SCE_SAVEDATA_DIALOG_SYSMMSG_TYPE_NOSPACE</code>	4	Displays "There is not enough free space on the memory card. To continue using the application, you must create at least XXXX MB of free space. Press the PS button to pause this application, and then delete other applications or content."
<code>SCE_SAVEDATA_DIALOG_SYSMMSG_TYPE_PROGRESS</code>	5	Displays "Saving... Do not power off the system or close the application.", "Loading..." or "Please wait..."
<code>SCE_SAVEDATA_DIALOG_SYSMMSG_TYPE_FINISHED</code>	6	Displays "Saving complete.", "Loading complete." or "Deletion complete."
<code>SCE_SAVEDATA_DIALOG_SYSMMSG_TYPE_CONFIRM_CANCEL</code>	7	Displays "Do you want to cancel saving?", "Do you want to cancel loading?" or "Do you want to cancel deleting?"
<code>SCE_SAVEDATA_DIALOG_SYSMMSG_TYPE_FILE_CORRUPTED</code>	8	Displays "The file is corrupt."

SCE CONFIDENTIAL

Value	(Number)	Description
SCE_SAVEDATA_DIALOG_SYSMMSG_TYPE_NOSPACE_CONTINUABLE	9	Displays "Could not save the file. There is not enough free space on the memory card. To save your progress in the application, you must create at least XXXX KB of free space. To create the free space, press PS button to pause this application, and then delete other applications or content."
SCE_SAVEDATA_DIALOG_SYSMMSG_TYPE_NODATA_IMPORT	12	Displays "There is no saved data that can be used with this application. If the saved data you want to use is on a PS Vita card, select the import saved data icon on the LiveArea™ screen."

Save Data Dialog for which SCE_SAVEDATA_DIALOG_SYSMMSG_TYPE_PROGRESS has been specified for *sysMsgType* cannot be closed by user operation. It must be closed from the caller side by using `sceSaveDataDialogSubClose()` at an arbitrary timing.

If SCE_SAVEDATA_DIALOG_SYSMMSG_TYPE_NOSPACE and SCE_SAVEDATA_DIALOG_SYSMMSG_TYPE_NOSPACE_CONTINUABLE are specified in *sysMsgType*, the insufficient space requested by the last insufficient file system free space error occurring will be formatted into appropriate units and displayed in the message. The application cannot specify the insufficient space displayed. If the application's save data drive (savedata0:) is mounted to the PlayStation®Vita card, an error notification ("An error has occurred. (C2-13322-1, 0x80100aa0)") will be displayed instead of a message.

File system free space will run out when saving save data with the application utility library, or when saving a database with libSceSqlite; therefore, display this message when file system free space runs out with these libraries.

Specify 0 in *value*.

The save data specified in *targetSlot* is displayed on the screen along with thumbnails and other information. For details on the specification of the save data to be displayed, refer to the "Specification of Save Data" chapter.

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

See Also

`sceSaveDataDialogInit()`, `sceSaveDataDialogSubClose()`
`SceSaveDataDialogParam`, `SceSaveDataDialogProgressBarParam`

SCE CONFIDENTIAL

SceSaveDataDialogErrorCodeParam

Structure for error code display

Definition

```
#include <savedata_dialog.h>
typedef struct SceSaveDataDialogErrorCodeParam {
    SceInt32 errorCode;
    SceAppUtilSaveDataSlot targetSlot;
    SceChar8 reserved[32];
} SceSaveDataDialogErrorCodeParam;
```

Members

errorCode Specifies the error code to be displayed
targetSlot Specifies the save data slot ID of the display target save data (details below)
reserved Reserved area (fill with all 0s)

Description

This structure is used to perform the setting for the error code display. It is used when calling Save Data Dialog in the `SCE_SAVEDATA_DIALOG_MODE_ERROR_CODE` mode. Fill it with all 0s during initialization.

Specify the error values returned by the various library APIs in *errorCode*. The error values are converted into the appropriate expression and displayed. The displayed dialog has one button, the **OK** button. The message to be displayed will automatically switch between saving, loading, etc. based on the value set for `SceSaveDataDialogParam.dispType`.

The save data specified in *targetSlot* is displayed on the screen along with thumbnails and other information. For details on the specification of the save data to be displayed, refer to the "Specification of Save Data" chapter.

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

See Also

`sceSaveDataDialogInit()`, `SceSaveDataDialogParam`

SCE CONFIDENTIAL

SceSaveDataDialogProgressBarParam

Structure for progress bar display

Definition

```
#include <savedata_dialog.h>
typedef struct SceSaveDataDialogProgressBarParam {
    SceSaveDataDialogProgressBarType barType;
    SceSaveDataDialogSystemMessageParam sysMsgParam;
    const SceChar8 *msg;
    SceAppUtilSaveDataSlot targetSlot;
    SceChar8 reserved[32];
} SceSaveDataDialogProgressBarParam;
```

Members

barType Specifies the type of progress bar
sysMsgParam Specifies the type of message to be displayed (details below)
msg Arbitrary character string to be displayed on the screen (NULL termination, UTF-8)
targetSlot Specifies the save data slot ID of the display target save data (details below)
reserved Reserved area (fill with all 0s)

Description

This structure is used to perform the setting for the progress bar display. It is used when calling Save Data Dialog in the SCE_SAVEDATA_DIALOG_MODE_PROGRESS_BAR mode. Fill it with all 0s during initialization.

Save Data Dialog for which SCE_SAVEDATA_DIALOG_MODE_PROGRESS_BAR has been specified cannot be closed by user operation. It must be closed from the caller side by using `sceSaveDataDialogSubClose()` at an arbitrary timing.

The following value is input in *barType*.

Value	(Number)	Description
SCE_SAVEDATA_DIALOG_PROGRESSBAR_TYPE_PERCENTAGE	0	Display progress bar expressed as a percentage

Specify *sysMsgParam* to display a system defined message at the same time as the progress bar. For details on the structure itself, refer to the `SceSaveDataDialogSystemMessageParam` section.

The following value can be specified in *sysMsgParam.sysMsgType*.

Value	(Number)	Description
SCE_SAVEDATA_DIALOG_SYMSG_TYPE_PROGRESS	5	Display "Saving/Loading... Please wait."

Specify *msg* to display a user specified character string at the same time as the progress bar. Specify the character string to be displayed on the screen with UTF-8. The character string termination must be NULL. Up to three line breaks can be inserted and the maximum size is SCE_SAVEDATA_DIALOG_USER_MSG_SIZE. When a string exceeding the maximum size is specified, the exceeding section will be cut off.

**sysMsgParam* and *msg* cannot be specified simultaneously. If a value is specified in *sysMsgParam.sysMsgType*, NULL must be specified in *msg*. Also, if no value is set in *sysMsgParam.sysMsgType*, a parameter error occurs even if NULL was set to *msg*.

SCE CONFIDENTIAL

The save data specified in *targetSlot* is displayed on the screen along with thumbnails and other information. For details on the specification of the save data to be displayed, refer to the "Specification of Save Data" chapter.

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

See Also

`sceSaveDataDialogInit()`, `sceSaveDataDialogSubClose()`

`SceSaveDataDialogParam`, `SCE_SAVEDATA_DIALOG_USER_MSG_SIZE`

000004892117

SCE CONFIDENTIAL

SceSaveDataDialogSlotConfigParam

Structure for save data slot setting

Definition

```
#include <savedata_dialog.h>
typedef struct SceSaveDataDialogSlotConfigParam {
    const SceAppUtilMountPoint *mountPoint;
    const SceSaveDataDialogAppSubDirParam *appSubDir;
    SceChar8 reserved[32];
} SceSaveDataDialogSlotConfigParam;
```

Members

mountPoint Specifies the location of the save data slot for displaying (details below)
appSubDir Specifies subdirectories under app: (details below)
reserved Reserved area (fill all with zeroes)

Description

This structure performs save data slot settings. It sets the interpretation of all save data slot parameters specified when Save Data Dialog is called. Fill all with zeroes at the initialization.

Save Data Dialog processing is performed for the save data slots present on the save data mount point specified in *mountPoint*. Either "savedata0:" or "savedata1:" can be specified. When NULL is specified, the function will behave as if the "savedata0:" have been specified.

appSubDir can specify the subdirectories under app0: for the file path specified in the *SceAppUtilSaveDataSlot* structure. If specification is not necessary, specify NULL. For details, refer to the *SceSaveDataDialogAppSubDirParam* section.

reserved is a reserved area for future function expansion. It must be filled with zeroes.

See Also

SceSaveDataDialogAppSubDirParam, *SceAppUtilMountPoint*, *sceSaveDataDialogInit()*

SCE CONFIDENTIAL

SceSaveDataDialogAppSubDirParam

Subdirectory Setting Structure

Definition

```
#include <savedata_dialog.h>
typedef struct SceSaveDataDialogAppSubDirParam {
    SceChar8 srcDir[SCE_APPUTIL_SAVEDATA_SLOT_ICON_PATH_MAXSIZE];
    SceChar8 dstDir[SCE_APPUTIL_SAVEDATA_SLOT_ICON_PATH_MAXSIZE];
    SceChar8 reserved[32];
} SceSaveDataDialogAppSubDirParam;
```

Members

srcDir Source path
dstDir Destination path
reserved Reserved area (fill all with zeroes)

Description

This structure is used to set the subdirectory under app0: for the file path specified by the `SceAppUtilSaveDataSlot` structure.

For *srcDir*, specify the character string of the file path specified by the `SceAppUtilSaveDataSlot` structure that is to be replaced. For *dstDir*, specify the character string that is to be replaced with.

Example) in the case of *srcDir*: oldDir and *dstDir*: newDir, path conversion is performed as follows

app0:oldDir/icon0.png -> app0:newDir/icon0.png

reserved is a reserved area for future function expansion. It must be filled with zeroes.

See Also

`SceSaveDataDialogSlotConfigParam`, `sceSaveDataDialogInit()`

Obtaining Operation Status

SCE CONFIDENTIAL

sceSaveDataDialogGetStatus

Get operation status of Save Data Dialog

Definition

```
#include <savedata_dialog.h>
SceCommonDialogStatus sceSaveDataDialogGetStatus()
```

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	Save Data Dialog is not running
SCE_COMMON_DIALOG_STATUS_RUNNING	0x1	Save Data Dialog is running
SCE_COMMON_DIALOG_STATUS_FINISHED	0x2	Save Data Dialog operation has finished

Description

This function gets the operation status of Save Data Dialog.

The default value of the operation status is SCE_COMMON_DIALOG_STATUS_NONE.

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

The operation status of Save Data Dialog does not change automatically to SCE_COMMON_DIALOG_STATUS_FINISHED until sceSaveDataDialogFinish() or sceSaveDataDialogAbort(), described later, is called.

sceSaveDataDialogTerm() can be called only while the operation status is SCE_COMMON_DIALOG_STATUS_FINISHED.

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

This function is multithread safe.

Examples

```
SceCommonDialogStatus stat = sceSaveDataDialogGetStatus();
```

See Also

sceSaveDataDialogInit(), sceSaveDataDialogSubClose(), sceSaveDataDialogAbort(), sceSaveDataDialogGetResult(), sceSaveDataDialogTerm()

SCE CONFIDENTIAL

sceSaveDataDialogGetSubStatus

Get operation substatus of Save Data Dialog

Definition

```
#include <savedata_dialog.h>
SceCommonDialogStatus sceSaveDataDialogGetSubStatus ()
```

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	Save Data Dialog is not running
SCE_COMMON_DIALOG_STATUS_RUNNING	0x1	Save Data Dialog is running
SCE_COMMON_DIALOG_STATUS_FINISHED	0x2	Save Data Dialog operation has finished

Description

This function gets the operation substatus of Save Data Dialog. For the substatus details, refer to the "Save Data Dialog Overview" document.

This function can be called only while the operation status is SCE_COMMON_DIALOG_STATUS_RUNNING. If this function is called at times other than the above, SCE_COMMON_DIALOG_STATUS_NONE is returned.

When calling sceSaveDataDialogInit(), sceSaveDataDialogContinue(), or sceSaveDataDialogFinish() is successful, the operation substatus immediately changes to SCE_COMMON_DIALOG_STATUS_RUNNING.

When Save Data Dialog is closed either through user operation or by calling sceSaveDataDialogSubClose(), the operation substatus changes to SCE_COMMON_DIALOG_STATUS_FINISHED after a while. sceSaveDataDialogGetResult(), sceSaveDataDialogContinue(), and sceSaveDataDialogFinish() can be called only while the operation status is SCE_COMMON_DIALOG_STATUS_FINISHED.

This function is multithread safe.

Examples

```
SceCommonDialogStatus stat = sceSaveDataDialogGetSubStatus();
```

See Also

```
sceSaveDataDialogInit(), sceSaveDataDialogSubClose(), sceSaveDataDialogAbort(),
sceSaveDataDialogGetResult(), sceSaveDataDialogContinue(),
sceSaveDataDialogFinish()
```


Updating Display Information

SCE CONFIDENTIAL

sceSaveDataDialogProgressBarInc

Increase progress bar rate

Definition

```
#include <savedata_dialog.h>
SceInt32 sceSaveDataDialogProgressBarInc (
    SceSaveDataDialogProgressBarTarget target,
    SceUInt32 delta
)
```

Arguments

target Target to be increased
delta Increase value

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_RUNNING	0x80020404	Called during other than appropriate time (details below)
SCE_COMMON_DIALOG_ERROR_NOT_SUPPORTED	0x80020405	Not support the current operation mode (details below)
SCE_SAVEDATA_DIALOG_ERROR_PARAM	0x80100b01	Parameter error (details below)
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

Description

This function increases the progress bar rate of Save Data Dialog displaying the progress bar. The label indicating the progress rate is updated according to the increasing progress rate, and the progress bar extends smoothly with an animated image.

This function can be called only when the Save Data Dialog operation mode is SCE_SAVEDATA_DIALOG_MODE_PROGRESS_BAR. If called during other operation modes, SCE_COMMON_DIALOG_ERROR_NOT_SUPPORTED is returned.

Also, this function can be called only when both the Save Data Dialog operation status and operation substatus are during the time of SCE_COMMON_DIALOG_STATUS_RUNNING. If called at other times, SCE_COMMON_DIALOG_ERROR_NOT_RUNNING is returned.

Call while these conditions are satisfied.

Refer to the sceSaveDataDialogGetStatus() section for details on operation status and the sceSaveDataDialogGetSubStatus() section for details on operation substatus.

Specify the target of which the progress rate is to be increased to *target*. The following value is set.

Value	(Number)	Description
SCE_SAVEDATA_DIALOG_PROGRESSBAR_TARGET_BAR_DEFAULT	0	Progress bar expressed as a percentage

SCE CONFIDENTIAL

`SCE_SAVEDATA_DIALOG_ERROR_PARAM` will be returned as the value of the function when values other than the above are specified.

Specify the increase value of the progress rate to *delta*. The initial value of the progress bar rate, which is acquired by specifying `SCE_SAVEDATA_DIALOG_PROGRESSBAR_TYPE_PERCENTAGE` to `SceSaveDataDialogProgressBarParam.barType`, is 0%. The percentage is increased by the value specified to *delta*, and the maximum value is 100%. Even if it is attempted to increase the progress rate after reaching 100%, the rate will not be changed.

This function is multithread safe.

Examples

```
// a thread different from the thread that has called sceSaveDataDialogInit()
for( SceUInt32 rate=0; rate<=100; rate+10 ) {
    sceSaveDataDialogProgressBarInc(
        SCE_SAVEDATA_DIALOG_PROGRESSBAR_TARGET_BAR_DEFAULT,
        10 );
}

sceSaveDataDialogSubClose();
```

See Also

`sceSaveDataDialogInit()`, `sceSaveDataDialogSubClose()`

SCE CONFIDENTIAL

sceSaveDataDialogProgressBarSetValue

Specify progress bar rate

Definition

```
#include <savedata_dialog.h>
SceInt32 sceSaveDataDialogProgressBarSetValue (
    SceSaveDataDialogProgressBarTarget target,
    SceUInt32 rate
)
```

Arguments

target Target to be specified
rate Value to be specified

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_RUNNING	0x80020404	Called during other than appropriate time (details below)
SCE_COMMON_DIALOG_ERROR_NOT_SUPPORTED	0x80020405	Not support the current operation mode (details below)
SCE_SAVEDATA_DIALOG_ERROR_PARAM	0x80100b01	Parameter error (details below)
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

Description

This function specifies the progress bar rate of Save Data Dialog displaying the progress bar. The label indicating the progress rate is updated according to the specified progress rate, and the length of the progress bar immediately changes to a suitable length without an animated image.

This function can be called only when the Save Data Dialog operation mode is SCE_SAVEDATA_DIALOG_MODE_PROGRESS_BAR. If called during other operation modes, SCE_COMMON_DIALOG_ERROR_NOT_SUPPORTED is returned.

Also, this function can be called only when both the Save Data Dialog operation status and operation substatus are during the time of SCE_COMMON_DIALOG_STATUS_RUNNING. If called at other times, SCE_COMMON_DIALOG_ERROR_NOT_RUNNING is returned.

Call while these conditions are satisfied.

Refer to the sceSaveDataDialogGetStatus() section for details on operation status and the sceSaveDataDialogGetSubStatus() section for details on operation substatus.

Specify the target of which the progress rate is to be specified to *target*. The following value is input.

Value	(Number)	Description
SCE_SAVEDATA_DIALOG_PROGRESSBAR_TARGET_BAR_DEFAULT	0	Progress bar expressed as a percentage

SCE CONFIDENTIAL

SCE_SAVEDATA_DIALOG_ERROR_PARAM will be returned as the value of the function when values other than the above are specified.

Specify the progress rate to be specified to *rate*. The maximum value of the progress bar rate, which is acquired by specifying SCE_SAVEDATA_DIALOG_PROGRESSBAR_TYPE_PERCENTAGE to SceSaveDataDialogProgressBarParam.*barType*, is 100%. If a value larger than this is specified, the value is automatically rounded to 100%.

This function is multithread safe.

Examples

```
SceSaveDataDialogParam param;
SceSaveDataDialogProgressBarParam progBarParam;

sceSaveDataDialogParamInit( &param );
sceClibMemset( &progBarParam, 0, sizeof(progBarParam) );
param.mode = SCE_SAVEDATA_DIALOG_MODE_PROGRESS_BAR;
param.progBarParam = &progBarParam;
param.progBarParam->sysMsgParam.sysMsgType =
SCE_SAVEDATA_DIALOG_SYSMMSG_TYPE_PROGRESS;

if ( sceSaveDataDialogInit( &param ) < 0 ) {
    // error
}

sceSaveDataDialogProgressBarSetValue(
    SCE_SAVEDATA_DIALOG_PROGRESSBAR_TARGET_BAR_DEFAULT,
    50 ); // Start from 50%
}
```

See Also

```
sceSaveDataDialogInit()
```

Closing Save Data Dialog

sceSaveDataDialogSubClose

Close Save Data Dialog

Definition

```
#include <savedata_dialog.h>
SceInt32 sceSaveDataDialogSubClose()
```

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_RUNNING	0x80020404	Called during other than appropriate operation substatus (details below)
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

Description

This function closes Save Data Dialog.

This function can be called only while the operation substatus of Save Data Dialog is SCE_COMMON_DIALOG_STATUS_RUNNING. If this function is called at times other than the above, SCE_COMMON_DIALOG_ERROR_NOT_RUNNING is returned.

If calling this function is successful, the operation status changes to SCE_COMMON_DIALOG_STATUS_FINISHED after the finish processing.

Note that these are not operation statuses.

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

When Save Data Dialog is closed with this function, calling sceSaveDataDialogGetResult() returns the following.

```
SceSaveDataDialogResult.result      : 0
SceSaveDataDialogResult.buttonId   : SCE_SAVEDATA_DIALOG_BUTTON_ID_INVALID
```

Since a user specified message dialog for which SCE_SAVEDATA_DIALOG_BUTTON_TYPE_NONE was specified as the button type, or some of system defined message dialog without the buttons and the progress bar dialog, cannot be closed through user operation, they must be closed with this function at an arbitrary timing. This function can be used also for other Save Data Dialogs that can be closed through user operation.

This function is multithread safe.

SCE CONFIDENTIAL

Examples

```
SceCommonDialogStatus stat;

while(1) {
    stat = sceSaveDataDialogGetStatus();
    if( stat == SCE_COMMON_DIALOG_STATUS_RUNNING ) {
        stat = sceSaveDataDialogGetSubStatus();
        if( stat == SCE_COMMON_DIALOG_STATUS_RUNNING && need_close ) {
            sceSaveDataDialogSubClose();
            break;
        }
    }
    else if( stat == SCE_COMMON_DIALOG_STATUS_FINISHED ) {
        sceSaveDataDialogTerm();
        break;
    }
}
```

See Also

```
sceSaveDataDialogGetSubStatus(), sceSaveDataDialogGetResult()
```


Retrieval of Save Data Dialog Call Result

SCE CONFIDENTIAL

sceSaveDataDialogGetResult

Get call result of Save Data Dialog

Definition

```
#include <savedata_dialog.h>
SceInt32 sceSaveDataDialogGetResult(
    SceSaveDataDialogResult *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 <i>result</i>
SCE_COMMON_DIALOG_ERROR_NOT_FINISHED	0x80020410	Called during other than the appropriate operation status (details below)
SCE_SAVEDATA_DIALOG_ERROR_PARAM	0x80100b01	Parameter error (details below)
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

Description

This function retrieves the call result of Save Data Dialog.

This function can be called only while the operation status or the operation substatus of Save Data 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 sceSaveDataDialogGetStatus() section, and for details on the operation substatuses, refer to the sceSaveDataDialogGetSubStatus() section.

The call result of Save Data Dialog is stored in *result*. For details on the call results, refer to the SceSaveDataDialogResult section.

Be sure to initialize the argument *result* before passing it to this function.

If SceSaveDataDialogResult.reserved is not filled with 0s, SCE_SAVEDATA_DIALOG_ERROR_PARAM is returned as the value of the function.

This function is multithread safe.

SCE CONFIDENTIAL

Examples

```
SceCommonDialogStatus stat;

while(1) {
    stat = sceSaveDataDialogGetStatus();
    else if( stat == SCE_COMMON_DIALOG_STATUS_FINISHED ) {
        SceSaveDataDialogResult result;
        sceClibMemset( &result, 0, sizeof(result) );
        if( 0 > sceSaveDataDialogGetResult( &result ) ) {
            // error
        }
        sceSaveDataDialogTerm();
        break;
    }
}
```

See Also

SceSaveDataDialogResult, sceSaveDataDialogGetStatus(),
sceSaveDataDialogGetSubStatus()

SCE CONFIDENTIAL

SceSaveDataDialogResult

Structure for retrieving Save Data Dialog call result

Definition

```
#include <savedata_dialog.h>
typedef struct SceSaveDataDialogResult {
    SceSaveDataDialogMode mode;
    SceInt32 result;
    SceSaveDataDialogButtonId buttonId;
    SceAppUtilSaveDataSlotId slotId;
    SceSaveDataDialogSlotInfo *slotInfo;
    void *userdata;
    SceChar8 reserved[32];
} SceSaveDataDialogResult;
```

Members

mode Stores *mode* at the time of call

result Stores the call result (details below)

buttonId Stores the ID of the button selected by the user (details below)

slotId Stores the save data slot ID of the save data that is displayed or selected (details below)

slotInfo Stores the detailed information of the save data indicated by *slotId*.
If not required, NULL specification is possible

userdata User specified data (details below)

reserved Reserved area (fill with all 0s)

Description

This structure receives the Save Data Dialog call result. It is passed to `sceSaveDataDialogGetResult()`. Fill it with all 0s during initialization.

The value of `SceSaveDataDialogParam.mode` specified with `sceSaveDataDialogInit()` or `sceSaveDataDialogContinue()` is stored. It can be used to determine which function's calling result it is.

The call result of Save Data 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 selected button. Closed with <code>sceSaveDataDialogSubClose()</code>
SCE_COMMON_DIALOG_RESULT_USER_CANCELED	0x1	User performed cancel operation.
SCE_COMMON_DIALOG_RESULT_ABORTED	0x2	Abortion through <code>sceSaveDataDialogAbort()</code>

In the case of an error, one of the following error codes (negative value) is stored.

Value	(Number)	Description
SCE_SAVEDATA_DIALOG_ERROR_PARAM	0x80100b01	Parameter error
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

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

Value	(Number)	Description
SCE_SAVEDATA_DIALOG_BUTTON_ID_INVALID	0	No button was selected
SCE_SAVEDATA_DIALOG_BUTTON_ID_OK	1	The OK button was selected
SCE_SAVEDATA_DIALOG_BUTTON_ID_YES	1	The Yes button was selected
SCE_SAVEDATA_DIALOG_BUTTON_ID_NO	2	The No button was selected

The following save data slot IDs of the display target save data specified with `sceSaveDataDialogInit()` are stored in *slotId*.

- `SceSaveDataDialogFixedParam.targetSlot.id`
- `SceSaveDataDialogUserMessageParam.targetSlot.id`
- `SceSaveDataDialogSystemMessageParam.targetSlot.id`
- `SceSaveDataDialogErrorCodeParam.targetSlot.id`
- `SceSaveDataDialogProgressBarParam.targetSlot.id`

The ID of the save data selected by the user is stored only when the save data list display function is called. However, if the list has been closed with a cancel operation, `SCE_APPUTIL_SAVEDATA_SLOT_ID_INVALID(0xffffffff)` is returned.

The detailed information of the save data can be additionally retrieved by specifying the address of the structure in *slotInfo*. For details, refer to the `SceSaveDataDialogSlotInfo` section. This can also be omitted by specifying NULL.

The value of `SceSaveDataDialogParam.userdata` specified with `sceSaveDataDialogInit()` or `sceSaveDataDialogContinue()` is stored in *userdata*.

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

The following are typical call result examples.

User selected the OK button

```
result      : 0
buttonId    : SCE_SAVEDATA_DIALOG_BUTTON_ID_OK
```

User selected the Yes or the No button

```
result      : 0
buttonId    : SCE_SAVEDATA_DIALOG_BUTTON_ID_YES /
              SCE_SAVEDATA_DIALOG_BUTTON_ID_NO
```

User selected the Cancel button (Cross button on the top left of the list display screen)

```
result      : SCE_COMMON_DIALOG_RESULT_USER_CANCELED
buttonId    : SCE_SAVEDATA_DIALOG_BUTTON_ID_INVALID
```

Save Data Dialog was closed with `sceSaveDataDialogSubClose()`

```
result      : 0
buttonId    : SCE_SAVEDATA_DIALOG_BUTTON_ID_INVALID
```

Save Data Dialog was closed with `sceSaveDataDialogAbort()`

```
result      : SCE_COMMON_DIALOG_RESULT_ABORTED
buttonId    : SCE_SAVEDATA_DIALOG_BUTTON_ID_INVALID
```

See Also

`sceSaveDataDialogInit()`, `sceSaveDataDialogContinue()`,
`sceSaveDataDialogSubClose()`, `sceSaveDataDialogAbort()`,
`sceSaveDataDialogGetResult()`, `SceSaveDataDialogSlotInfo`

SCE CONFIDENTIAL

SceSaveDataDialogSlotInfo

Save data slot information structure

Definition

```
#include <savedata_dialog.h>
typedef struct SceSaveDataDialogSlotInfo {
    SceUInt32 isExist;
    SceAppUtilSaveDataSlotParam *slotParam;
    SceChar8 reserved[32];
} SceSaveDataDialogSlotInfo;
```

Members

isExist Information on whether the save data slot exists or not (0: no 1: yes)

slotParam Stores parameter of the save data slot
If not required, NULL specification is possible.

reserved Reserved area (fill with all 0s)

Description

This structure stores save data slot information.

Fill it with all 0s during initialization.

One of following is returned to *isExist*

Value	Description
0	Save data slot does not exist
1	Save data slot exists

By specifying address to the entity of `SceAppUtilSaveDataSlotParam` structure to *slotParam*, system file parameter of the save data can be retrieved additionally.

For details on the `SceAppUtilSaveDataSlotParam` structure, refer to the "Application Utility Reference" document. This can also be omitted by specifying NULL.

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

If *reserved* is not filled with all 0s, a parameter error will occur, and

`SCE_SAVEDATA_DIALOG_ERROR_PARAM` will return to `SceSaveDataDialogResult.result`, which is obtained with `sceSaveDataDialogGetResult()`.

See Also

`sceSaveDataDialogGetResult()`,
`SceSaveDataDialogResult`, `SceAppUtilSaveDataSlotParam`

Continuous Call of Save Data Dialog

SCE CONFIDENTIAL

sceSaveDataDialogContinue

Continuously call Save Data Dialog

Definition

```
#include <savedata_dialog.h>
SceInt32 sceSaveDataDialogContinue (
    const SceSaveDataDialogParam *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_NULL	0x80020402	NULL was specified for the argument <i>param</i>
SCE_COMMON_DIALOG_ERROR_NOT_RUNNING	0x80020404	Called during other than appropriate operation status (details below)
SCE_COMMON_DIALOG_ERROR_NOT_FINISHED	0x80020410	Called during other than appropriate operation substatus (details below)
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

Description

This function continuously calls Save Data Dialog.

When this function is called while the operating status of Save Data Dialog is other than SCE_COMMON_DIALOG_STATUS_RUNNING, SCE_COMMON_DIALOG_ERROR_NOT_RUNNING is returned. Furthermore, when this function is called while the operation substatus is not SCE_COMMON_DIALOG_STATUS_FINISHED, SCE_COMMON_DIALOG_ERROR_NOT_FINISHED is returned.

Call this function while both conditions are satisfied.

For details on the operation statuses, refer to the `sceSaveDataDialogGetStatus()` section, and for details on the operation substatuses, refer to the `sceSaveDataDialogGetSubStatus()` section.

The *param* specification method is the same as for `sceSaveDataDialogInit()`. This function continues to call a different Save Data Dialog function while maintaining the display of the save data on the screen.

If some of the *param* settings (see list below) are omitted (0 is specified) when calling `sceSaveDataDialogContinue()`, the value specified with `sceSaveDataDialogInit()` is inherited.

- *dispType*

Also, the following setting will be ignored:

- *commonParam*

This function is multithread safe.

SCE CONFIDENTIAL

Examples

```
SceCommonDialogStatus stat;

while(1) {
    stat = sceSaveDataDialogGetStatus();
    if( stat == SCE_COMMON_DIALOG_STATUS_FINISHED ) {
        SceSaveDataDialogParam param;
        SceSaveDataDialogErrorCodeParam errorCodeParam;
        sceSaveDataDialogParamInit( &param );
        sceClibMemset( &errorCodeParam, 0, sizeof(errorCodeParam) );
        param.mode = SCE_SAVEDATA_DIALOG_MODE_ERROR_CODE;
        param.targetSlot = 0x01;
        param.errorCodeParam = &errorCodeParam;

        // set error code returned by library
        param.errorCodeParam->errorCode = ret;
        if ( sceSaveDataDialogContinue( &param ) < 0 ) {
            //error
        }
    }
}
```

See Also

SceSaveDataDialogParam, sceSaveDataDialogInit(), sceSaveDataDialogGetStatus(),
sceSaveDataDialogGetSubStatus()

SCE CONFIDENTIAL

sceSaveDataDialogFinish

Finish continuous call of Save Data Dialog

Definition

```
#include <savedata_dialog.h>
SceInt32 sceSaveDataDialogFinish(
    const SceSaveDataDialogFinishParam *finishParam
)
```

Arguments

finishParam Finish processing setting 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_NULL	0x80020402	NULL was specified in the argument <i>finishParam</i>
SCE_COMMON_DIALOG_ERROR_NOT_RUNNING	0x80020404	Called during other than the appropriate operation status (details below)
SCE_COMMON_DIALOG_ERROR_NOT_FINISHED	0x80020410	Called during other than the appropriate operation substatus (details below)
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

Description

This function ends calling of Save Data Dialog.

Even when Save Data Dialog is closed through user operation or `sceSaveDataDialogSubClose()`, the operation substatus simply changes to `SCE_COMMON_DIALOG_STATUS_FINISHED` and the operation status remains `SCE_COMMON_DIALOG_STATUS_RUNNING`. The operation status changes to `SCE_COMMON_DIALOG_STATUS_FINISHED` when this function is called.

This is because, each time one function is called, a choice is made between continuing the save flow (`sceSaveDataDialogContinue()`) or finishing it (`sceSaveDataDialogFinish()`).

When this function is called while the operation status of Save Data Dialog is other than `SCE_COMMON_DIALOG_STATUS_RUNNING`, `SCE_COMMON_DIALOG_ERROR_NOT_RUNNING` is returned. Furthermore, when this function is called while the operation substatus is not `SCE_COMMON_DIALOG_STATUS_FINISHED`, `SCE_COMMON_DIALOG_ERROR_NOT_FINISHED` is returned.

Call this function while both conditions are satisfied.

For details on the operation statuses, refer to the `sceSaveDataDialogGetStatus()` section, and for details on the operation substatuses, refer to the `sceSaveDataDialogGetSubStatus()` section.

Specify the Save Data Dialog finish processing in *finishParam*. For details, refer to the `SceSaveDataDialogFinishParam` section.

This function is multithread safe.

SCE CONFIDENTIAL

Examples

```
SceCommonDialogStatus stat;

while(1) {
    stat = sceSaveDataDialogGetStatus();
    if( stat == SCE_COMMON_DIALOG_STATUS_FINISHED ) {
        SceSaveDataDialogFinishParam finishParam;
        sceClibMemset( &finishParam, 0, sizeof(finishParam) );
        finishParam.flag = SCE_SAVEDATA_DIALOG_FINISH_FLAG_DEFAULT;

        if ( sceSaveDataDialogFinish( &finishParam ) < 0 ) {
            //error
        }
    }
}
```

See Also

SceSaveDataDialogFinishParam, sceSaveDataDialogGetStatus(),
sceSaveDataDialogGetSubStatus(), sceSaveDataDialogSubClose(),
sceSaveDataDialogContinue()

SCE CONFIDENTIAL

SceSaveDataDialogFinishParam

Structure for Save Data Dialog finish processing setting

Definition

```
#include <savedata_dialog.h>
typedef struct SceSaveDataDialogFinishParam {
    SceSaveDataDialogFinishFlag flag;
    SceChar8 reserved[32];
} SceSaveDataDialogFinishParam;
```

Members

flag Processing setting flag
reserved Reserved area (fill with all 0s)

Description

This structure sets the finish processing of Save Data Dialog. It is passed to `sceSaveDataDialogFinish()`. Fill all with zeroes at the initialization.

The following value is input in *flag*.

Value	(Number)	Description
SCE_SAVEDATA_DIALOG_FINISH_FLAG_DEFAULT	0	Default processing

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

See Also

`sceSaveDataDialogFinish()`

Aborting Save Data Dialog

SCE CONFIDENTIAL

sceSaveDataDialogAbort

Abort call of Save Data Dialog

Definition

```
#include <savedata_dialog.h>
SceInt32 sceSaveDataDialogAbort()
```

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	sceSaveDataDialogInit() is not called
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

Description

This function aborts calling of Save Data Dialog.

It can be called at any time between when sceSaveDataDialogInit() is called and sceSaveDataDialogTerm() 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, Save Data 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 sceSaveDataDialogGetStatus() section.

When Save Data Dialog is closed with this function, calling sceSaveDataDialogGetResult() returns the following.

```
SceSaveDataDialogResult.result : SCE_COMMON_DIALOG_RESULT_ABORTED
SceSaveDataDialogResult.buttonId : SCE_SAVEDATA_DIALOG_BUTTON_ID_INVALID
```

sceSaveDataDialogAbort() is used to promptly abort the Save Data Dialog display, for example when an urgent interrupt must be processed. Since the finish processing may be skipped, use sceSaveDataDialogSubClose() when wishing to close Save Data Dialog normally from the caller side.

This function is multithread safe.

Examples

```
SceCommonDialogStatus stat;

while(1) {
    stat = sceSaveDataDialogGetStatus();
    if( stat == SCE_COMMON_DIALOG_STATUS_RUNNING ) {
        if( need_abort ) {
            sceSaveDataDialogAbort();
            break;
        }
    }
    else if( stat == SCE_COMMON_DIALOG_STATUS_FINISHED ) {
        sceSaveDataDialogTerm();
        break;
    }
}
```

See Also

sceSaveDataDialogSubClose(), sceSaveDataDialogGetResult()

Terminating Save Data Dialog

SCE CONFIDENTIAL

sceSaveDataDialogTerm

End call of Save Data Dialog

Definition

```
#include <savedata_dialog.h>
SceInt32 sceSaveDataDialogTerm()
```

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	sceSaveDataDialogInit() is not called
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

Description

This function ends calling of Save Data Dialog. Calling must be ended with this function after Save Data Dialog has been called with sceSaveDataDialogInit().

This function can be called only while the operation status of Save Data 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 Save Data 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 sceSaveDataDialogGetStatus() section.

This function is multithread safe.

Examples

```
SceCommonDialogStatus stat;

while(1) {
    stat = sceSaveDataDialogGetStatus();
    if( stat == SCE_COMMON_DIALOG_STATUS_FINISHED ) {
        sceSaveDataDialogTerm();
        break;
    }
}
```

SCE CONFIDENTIAL

See Also

`sceSaveDataDialogInit()`, `sceSaveDataDialogGetStatus()`

000004892117

Save Data Dialog Call for Save Data for PSP™

SCE CONFIDENTIAL

scePspSaveDataDialogParamInit

Macro for call parameter initialization

Definition

```
#include <savedata_psp_dialog.h>
static inline
void scePspSaveDataDialogParamInit(ScePspSaveDataDialogParam *param)
{
    sceClibMemset( param, 0x0, sizeof(ScePspSaveDataDialogParam) );
    _sceCommonDialogSetMagicNumber( &param->commonParam );
    param->sdkVersion = SCE_PSP2_SDK_VERSION;
}
```

Arguments

param Call parameter

Return Values

None

Description

This is a macro function for initializing the Save Data Dialog call parameter for save data for PSP™ (PlayStation®Portable).

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

```
ScePspSaveDataDialogParam param;
ScePspSaveDataDialogSystemMessageParam sysMsgParam;

scePspSaveDataDialogParamInit( &param );
sceClibMemset( &sysMsgParam, 0, sizeof(sysMsgParam) );
param.mode = SCE_SAVEDATA_DIALOG_MODE_SYSTEM_MSG;
param.sysMsgParam = &sysMsgParam;
param.sysMsgParam->sysMsgType = SCE_SAVEDATA_DIALOG_SYMSG_TYPE_CONFIRM;

if ( scePspSaveDataDialogInit( &param ) < 0 ) {
    // error
}
```

See Also

ScePspSaveDataDialogParam

SCE CONFIDENTIAL

scePspSaveDataDialogInit

Call various features of Save Data Dialog for save data for PSP™

Definition

```
#include <savedata_psp_dialog.h>
SceInt32 scePspSaveDataDialogInit(
    const ScePspSaveDataDialogParam *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 function is already being called (details below)
SCE_COMMON_DIALOG_ERROR_NULL	0x80020402	NULL was specified for the <i>param</i> argument
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

Description

This is a version of `sceSaveDataDialogInit()` for save data for PSP™. Its basic specifications are the same as `sceSaveDataDialogInit()`.

Be sure to set values for *param* after performing initialization with the `scePspSaveDataDialogParamInit()` macro.

For the processing after Save Data Dialog has been called with this function for the display features for save data for PSP™, use the following functions applicable to the display features for save data for PSP™.

- `scePspSaveDataDialogContinue()`
- `scePspSaveDataDialogGetResult()`
- `scePspSaveDataDialogTerm()`

The following functions can be used for both the standard Save Data Dialog features and the display features for save data for PSP™.

- `sceSaveDataDialogGetStatus()`
- `sceSaveDataDialogGetSubStatus()`
- `sceSaveDataDialogProgressBarInc()`
- `sceSaveDataDialogProgressBarSetValue()`
- `sceSaveDataDialogSubClose()`
- `sceSaveDataDialogFinish()`
- `sceSaveDataDialogAbort()`

SCE CONFIDENTIAL

Examples

```
ScePspSaveDataDialogParam param;
ScePspSaveDataDialogUserMessageParam userMsgParam;
static const SceChar8 msg_hello[] = "Hello world!";

scePspSaveDataDialogParamInit( &param );
sceClibMemset( &userMsgParam, 0, sizeof(userMsgParam) );
param.mode = SCE_SAVEDATA_DIALOG_MODE_USER_MSG;
param.userMsgParam = &userMsgParam;
param.userMsgParam->msg = msg_hello;
param.userMsgParam->buttonType = SCE_SAVEDATA_DIALOG_BUTTON_TYPE_OK;

if ( scePspSaveDataDialogInit( &param ) < 0 ) {
    //error
}
```

See Also

ScePspSaveDataDialogParam, scePspSaveDataDialogParamInit(),
scePspSaveDataDialogContinue(), scePspSaveDataDialogGetResult(),
scePspSaveDataDialogTerm()

SCE CONFIDENTIAL

ScePspSaveDataDialogParam

Call structure for Save Data Dialog for save data for PSP™

Definition

```
#include <savedata_psp_dialog.h>
typedef struct ScePspSaveDataDialogParam {
    SceUInt32 sdkVersion;
    SceCommonDialogParam commonParam;
    SceSaveDataDialogMode mode;
    ScePspSaveDataDialogType dispType;
    ScePspSaveDataDialogFixedParam *fixedParam;
    ScePspSaveDataDialogListParam *listParam;
    ScePspSaveDataDialogUserMessageParam *userMsgParam;
    ScePspSaveDataDialogSystemMessageParam *sysMsgParam;
    ScePspSaveDataDialogErrorCodeParam *errorCodeParam;
    ScePspSaveDataDialogProgressBarParam *progBarParam;
    ScePspSaveDataDialogEnvFlag flag;
    void *userdata;
    SceChar8 reserved[32];
} ScePspSaveDataDialogParam;
```

Members

<i>sdkVersion</i>	SDK version
<i>commonParam</i>	Common parameters
<i>mode</i>	Operation mode
<i>dispType</i>	Display wording type
<i>fixedParam</i>	Save data fixed display parameter
<i>listParam</i>	Save data list display parameter
<i>userMsgParam</i>	User specified message display parameter
<i>sysMsgParam</i>	System defined message display parameter
<i>errorCodeParam</i>	Error code display parameter
<i>progBarParam</i>	Progress bar display parameter
<i>flag</i>	Environmental setting flag
<i>userdata</i>	User specified data
<i>reserved</i>	Reserved area (fill with all 0s)

Description

This is a version of the `SceSaveDataDialogParam` structure for save data for PSP™. The various member variables have been changed for save data for PSP™, but the basic specifications are the same as `SceSaveDataDialogParam`. For initialization, use `scePspSaveDataDialogParamInit()`.

Specify the display wording type of Save Data Dialog in *dispType*. One of the following values is input.

Value	(Number)	Description
<code>SCE_PSP_SAVEDATA_DIALOG_TYPE_LOAD</code>	1	Load dialog display for save data for PSP™

SCE CONFIDENTIAL

See Also

scePspSaveDataDialogParamInit(), scePspSaveDataDialogGetResult(),
ScePspSaveDataDialogFixedParam, ScePspSaveDataDialogUserMessageParam,
ScePspSaveDataDialogSystemMessageParam, ScePspSaveDataDialogErrorCodeParam,
ScePspSaveDataDialogProgressBarParam, ScePspSaveDataDialogResult

000004892117

SCE CONFIDENTIAL

ScePspSaveDataDialogFixedParam

Fixed save data display structure for save data for PSP™

Definition

```
#include <savedata_psp_dialog.h>
typedef struct ScePspSaveDataDialogFixedParam {
    SceAppUtilPspSaveDataDirName targetDirName;
    SceChar8 reserved[32];
} ScePspSaveDataDialogFixedParam;
```

Members

<i>targetDirName</i>	Specifies the target directory name of the save data for PSP™ to display
<i>reserved</i>	Reserved area (fill with all 0s)

Description

This is a version of *SceSaveDataDialogFixedParam* for save data for PSP™. Its basic specifications are the same as *SceSaveDataDialogFixedParam*.

The save data specified in *targetDirName* is displayed on the screen along with thumbnails and other information.

See Also

ScePspSaveDataDialogParam, *scePspSaveDataDialogInit()*

SCE CONFIDENTIAL

ScePspSaveDataDialogListParam

Save data list display structure for save data for PSP™

Definition

```
#include <savedata_psp_dialog.h>
typedef struct ScePspSaveDataDialogListParam {
    SceAppUtilPspSaveDataDirName *targetDirNameList;
    SceUInt32 dirNameListSize;
    ScePspSaveDataDialogFocusPos focusPos;
    SceAppUtilPspSaveDataDirName focusDirName;
    const SceChar8 *listTitle;
    ScePspSaveDataDialogListItemStyle itemStyle;
    SceChar8 reserved[32];
} ScePspSaveDataDialogListParam;
```

Members

<i>targetDirNameList</i>	Specifies the target list of the directory names of the save data for PSP™ for list display
<i>dirNameListSize</i>	Specifies the number of items of the list of save data specified in <i>targetDirNameList</i>
<i>focusPos</i>	Specifies the initial focus position on the list
<i>focusDirName</i>	Specifies the initial focus position when specifying with a directory name
<i>listTitle</i>	Specifies the title of the list
<i>itemStyle</i>	Specifies the display style of the listed item
<i>reserved</i>	Reserved area (fill with all 0s)

Description

This is a version of the `SceSaveDataDialogListParam` structure for save data for PSP™. Its basic specifications are the same as `SceSaveDataDialogListParam`.

The save data specified in *targetDirNameList* is displayed on the screen along with thumbnails and other information. The *targetDirNameList*'s referent must be held until `scePspSaveDataDialogTerm()` has been called, and the Save Data Dialog function call has been completed.

The number of units of save data specified in *dirNameListSize* is searched and displayed from the head of *targetDirNameList*. The maximum number that can be specified is defined by `SCE_SAVEDATA_DIALOG_SLOTLIST_MAXSIZE`.

For *focusDirName*, specify the focus target save data when `SCE_SAVEDATA_DIALOG_FOCUS_POS_ID` is specified to *focusPos*. If the specified save data is not saved in the list, the focus corresponds to the list head.

The title of the save data list to be displayed can be specified with UTF-8 character string in *listTitle*. String termination must be NULL. Line breaks cannot be inserted. The maximum size is `SCE_SAVEDATA_DIALOG_LIST_TITLE_SIZE`. If NULL is specified, a system default message will be displayed in accordance with the specified value of `ScePspSaveDataDialogParam.dispType`.

The display style of the listed item can be specified in *itemStyle*. The following values are input.

Value	(Number)	Description
SCE_PSP_SAVEDATA_DIALOG_LIST_ITEM_STYLE_TITLE_DATE_SUBTITLE	0	Performs display on three lines, in the order of Title/Date/Subtitle.
SCE_PSP_SAVEDATA_DIALOG_LIST_ITEM_STYLE_TITLE_SUBTITLE_DATE	1	Performs display on three lines, in the order of Title/Subtitle/Date.

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

See Also

ScePspSaveDataDialogParam, scePspSaveDataDialogInit(),
 SceAppUtilPspSaveDataDirName, SCE_SAVEDATA_DIALOG_LIST_TITLE_SIZE,
 SCE_SAVEDATA_DIALOG_SLOTLIST_MAXSIZE

SCE CONFIDENTIAL

ScePspSaveDataDialogUserMessageParam

User-specified message display structure for save data for PSP™

Definition

```
#include <savedata_psp_dialog.h>
typedef struct ScePspSaveDataDialogUserMessageParam {
    ScePspSaveDataDialogButtonType buttonType;
    const SceChar8 *msg;
    SceAppUtilPspSaveDataDirName targetDirName;
    SceChar8 reserved[32];
} ScePspSaveDataDialogUserMessageParam;
```

Members

<i>buttonType</i>	Specifies the type of button displayed on the dialog
<i>msg</i>	Arbitrary character string displayed on the screen (NULL termination, UTF-8)
<i>targetDirName</i>	Specifies the target directory name for the save data for PSP™ to display
<i>reserved</i>	Reserved area (fill with all 0s)

Description

This is a version of the `SceSaveDataDialogUserMessageParam` structure for save data for PSP™. Its basic specifications are the same as `SceSaveDataDialogUserMessageParam`.

The save data specified in *targetDirName* is displayed on the screen along with thumbnails and other information.

See Also

```
scePspSaveDataDialogInit(), sceSaveDataDialogSubClose(),
ScePspSaveDataDialogParam, SceAppUtilPspSaveDataDirName,
SCE_SAVEDATA_DIALOG_USER_MSG_SIZE
```

SCE CONFIDENTIAL

ScePspSaveDataDialogSystemMessageParam

System defined message display structure for save data for PSP™

Definition

```
#include <savedata_psp_dialog.h>
typedef struct ScePspSaveDataDialogSystemMessageParam {
    ScePspSaveDataDialogSystemMessageType sysMsgType;
    SceInt32 value;
    SceAppUtilPspSaveDataDirName targetDirName;
    SceChar8 reserved[32];
} ScePspSaveDataDialogSystemMessageParam;
```

Members

<i>sysMsgType</i>	Specifies the type of system defined message to be displayed
<i>value</i>	Specifies a value during a system defined message call requiring an additional value
<i>targetDirName</i>	Specifies the directory name for save data for PSP™ to display
<i>reserved</i>	Reserved area (fill with all 0s)

Description

This is a version of the `SceSaveDataDialogSystemMessageParam` structure for save data for PSP™. Its basic specifications are the same as `SceSaveDataDialogSystemMessageParam`.

The save data specified in *targetDirName* is displayed on the screen along with thumbnails and other information.

See Also

`scePspSaveDataDialogInit()`, `sceSaveDataDialogSubClose()`,
`ScePspSaveDataDialogParam`, `ScePspSaveDataDialogProgressBarParam`,
`ScePspSaveDataDialogParam`, `SceAppUtilPspSaveDataDirName`

SCE CONFIDENTIAL

ScePspSaveDataDialogErrorCodeParam

Error code display structure for save data for PSP™

Definition

```
#include <savedata_psp_dialog.h>
typedef struct ScePspSaveDataDialogErrorCodeParam {
    SceInt32 errorCode;
    SceAppUtilPspSaveDataDirName targetDirName;
    SceChar8 reserved[32];
} ScePspSaveDataDialogErrorCodeParam;
```

Members

<i>errorCode</i>	Specifies the error code to be displayed
<i>targetDirName</i>	Specifies the target directory name for the save data for PSP™ to display
<i>reserved</i>	Reserved area (fill with all 0s)

Description

This is a version of the `SceSaveDataDialogErrorCodeParam` structure for save data for PSP™. Its basic specifications are the same as `SceSaveDataDialogErrorCodeParam`.

The save data specified in *targetDirName* is displayed on the screen along with thumbnails and other information.

See Also

`scePspSaveDataDialogInit()`, `ScePspSaveDataDialogParam`,
`ScePspSaveDataDialogParam`, `SceAppUtilPspSaveDataDirName`

SCE CONFIDENTIAL

ScePspSaveDataDialogProgressBarParam

Progress bar display structure for save data for PSP™

Definition

```
#include <savedata_psp_dialog.h>
typedef struct ScePspSaveDataDialogProgressBarParam {
    ScePspSaveDataDialogProgressBarType barType;
    ScePspSaveDataDialogSystemMessageParam sysMsgParam;
    const SceChar8 *msg;
    SceAppUtilPspSaveDataDirName targetDirName;
    SceChar8 reserved[32];
} ScePspSaveDataDialogProgressBarParam;
```

Members

<i>barType</i>	Specifies the type of progress bar
<i>sysMsgParam</i>	Specifies the type of system defined message to be displayed
<i>msg</i>	Arbitrary character string to be displayed on the screen (NULL termination, UTF-8)
<i>targetDirName</i>	Specifies the target directory name for the save data for PSP™ to display
<i>reserved</i>	Reserved area (fill with all 0s)

Description

This is a version of the `SceSaveDataDialogProgressBarParam` structure for save data for PSP™. Its basic specifications are the same as `SceSaveDataDialogProgressBarParam`.

The save data specified in *targetDirName* is displayed on the screen along with thumbnails and other information.

See Also

```
scePspSaveDataDialogInit(), sceSaveDataDialogSubClose(),
ScePspSaveDataDialogParam, SceAppUtilPspSaveDataDirName,
SCE_SAVEDATA_DIALOG_USER_MSG_SIZE
```

Obtaining Save Data Dialog Call Result for Save Data for PSP™

SCE CONFIDENTIAL

scePspSaveDataDialogGetResult

Get Save Data Dialog call result for save data for PSP™

Definition

```
#include <savedata_psp_dialog.h>
SceInt32 scePspSaveDataDialogGetResult(
    ScePspSaveDataDialogResult *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 <i>result</i>
SCE_COMMON_DIALOG_ERROR_NOT_FINISHED	0x80020410	Called during other than the appropriate operation status
SCE_SAVEDATA_DIALOG_ERROR_PARAM	0x80100b01	Parameter error
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

Description

This is a version of `sceSaveDataDialogGetStatus()` for save data for PSP™. Its basic specifications are the same as `sceSaveDataDialogGetStatus()`.

After calling the display features for save data for PSP™ with `scePspSaveDataDialogInit()`, use this function to obtain the result.

Examples

```
SceCommonDialogStatus stat;

while(1) {
    stat = sceSaveDataDialogGetStatus();
    else if( stat == SCE_COMMON_DIALOG_STATUS_FINISHED ) {
        ScePspSaveDataDialogResult result;
        sceClibMemset( &result, 0, sizeof(result) );
        if( 0 > scePspSaveDataDialogGetResult( &result ) ) {
            // error
        }
        sceSaveDataDialogTerm();
        break;
    }
}
```

See Also

`ScePspSaveDataDialogResult`, `sceSaveDataDialogGetStatus()`,
`sceSaveDataDialogGetSubStatus()`

©SCEI

SCE CONFIDENTIAL

ScePspSaveDataDialogResult

Structure for obtaining Save Data Dialog call result for save data for PSP™

Definition

```
#include <savedata_psp_dialog.h>
typedef struct SceSaveDataDialogResult {
    SceSaveDataDialogMode mode;
    SceInt32 result;
    SceSaveDataDialogButtonId buttonId;
    SceAppUtilPspSaveDataDirName dirName;
    SceAppUtilPspSaveDataParamSfo *paramSfo;
    void *userdata;
    SceChar8 reserved[32];
} ScePspSaveDataDialogResult;
```

Members

<i>mode</i>	Stores <i>mode</i> at the time of call
<i>result</i>	Stores the call result
<i>buttonId</i>	Stores the ID of the button selected by the user
<i>dirName</i>	Stores the directory name of the save data that is displayed or selected
<i>paramSfo</i>	Stores system file (PARAM.SFO) information for the displayed or selected save data (currently unsupported)
<i>userdata</i>	User specified data
<i>reserved</i>	Reserved area (fill with all 0s)

Description

This is a version of the `SceSaveDataDialogResult` structure for save data for PSP™. Its basic specifications are the same as `SceSaveDataDialogResult`.

In *dirName*, the following target directory names specified with `scePspSaveDataDialogInit()` for the save data for PSP™ to display will be stored.

- `ScePspSaveDataDialogFixedParam.targetDirName`
- `ScePspSaveDataDialogUserMessageParam.targetDirName`
- `ScePspSaveDataDialogSystemMessageParam.targetDirName`
- `ScePspSaveDataDialogErrorCodeParam.targetDirName`
- `ScePspSaveDataDialogProgressBarParam.targetDirName`

The save data directory names for the save data selected by the user will be stored only when the save data list display feature for save data for PSP™ is called. However, if the list was closed with a cancel operation, an empty character string will be set.

For *paramSfo*, specify NULL.

See Also

`scePspSaveDataDialogInit()`, `scePspSaveDataDialogContinue()`,
`sceSaveDataDialogSubClose()`, `sceSaveDataDialogAbort()`,
`scePspSaveDataDialogGetResult()`

SCE CONFIDENTIAL

SceAppUtilPspSaveDataParamSfo

System file (PARAM.SFO) structure for save data for PSP™

Definition

```
#include <apputil/apputil_psp.h>
typedef struct SceAppUtilPspSaveDataParamSfo {
    SceChar8 saveDataDirectory
        [SCE_APPUTIL_PSP_SAVEDATA_PARAMSFO_DIRECTORY_SIZE];
    SceChar8 title
        [SCE_APPUTIL_PSP_SAVEDATA_PARAMSFO_TITLE_SIZE];
    SceChar8 saveDataTitle
        [SCE_APPUTIL_PSP_SAVEDATA_PARAMSFO_SD_TITLE_SIZE];
    SceChar8 detail
        [SCE_APPUTIL_PSP_SAVEDATA_PARAMSFO_DETAIL_SIZE];
    SceUChar8 parentalLev;
    SceChar8 reserved[3];
} SceAppUtilPspSaveDataParamSfo;
```

Members

<i>saveDataDirectory</i>	Directory name (NULL-terminated, ASCII)
<i>title</i>	Title (NULL-terminated, UTF-8)
<i>saveDataTitle</i>	Save data title (NULL-terminated, UTF-8)
<i>detail</i>	Detailed information (NULL-terminated, UTF-8)
<i>parentalLev</i>	Parental control level
<i>reserved</i>	Reserved area (fill with all 0s)

Description

This structure stores the system file (PARAM.SFO) information for save data for PSP™.

In *saveDataDirectory*, the directory name of the save data for PSP™ is stored.

In *title*, the game title name is stored.

In *saveDataTitle*, the save data title name is stored.

For *detail*, detailed information for the save data is specified.

In *parentalLev*, the parental control level for the save data is stored.

See Also

ScePspSaveDataDialogResult

Continuous Call of Save Data Dialog for Save Data for PSP™

SCE CONFIDENTIAL

scePspSaveDataDialogContinue

Continuously call Save Data Dialog for save data for PSP™

Definition

```
#include <savedata_psp_dialog.h>
SceInt32 scePspSaveDataDialogContinue (
    const ScePspSaveDataDialogParam *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_NULL	0x80020402	NULL was specified for the argument <i>param</i>
SCE_COMMON_DIALOG_ERROR_NOT_RUNNING	0x80020404	Called during other than appropriate operation status
SCE_COMMON_DIALOG_ERROR_NOT_FINISHED	0x80020410	Called during other than appropriate operation substatus
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

Description

This is a version of the `sceSaveDataDialogContinue()` for save data for PSP™. Its basic specifications are the same as `sceSaveDataDialogContinue()`.

The specification method for *param* is the same as with `scePspSaveDataDialogInit()`.

Examples

```
SceCommonDialogStatus stat;

while(1) {
    stat = sceSaveDataDialogGetStatus();
    if( stat == SCE_COMMON_DIALOG_STATUS_FINISHED ) {
        ScePspSaveDataDialogParam param;
        ScePspSaveDataDialogErrorCodeParam errorCodeParam;
        scePspSaveDataDialogParamInit( &param );
        sceClibMemset( &errorCodeParam, 0, sizeof(errorCodeParam) );
        param.mode = SCE_SAVEDATA_DIALOG_MODE_ERROR_CODE;
        param.targetSlot = 0x01;
        param.errorCodeParam = &errorCodeParam;

        // set error code returned by library
        param.errorCodeParam->errorCode = ret;
        if ( scePspSaveDataDialogContinue( &param ) < 0 ) {
            //error
        }
    }
}
```

©SCEI

SCE CONFIDENTIAL

```
}  
}
```

See Also

`ScePspSaveDataDialogParam`, `scePspSaveDataDialogInit()`,
`sceSaveDataDialogGetStatus()`, `sceSaveDataDialogGetSubStatus()`

000004892117

Terminating Save Data Dialog for Save Data for PSP™

SCE CONFIDENTIAL

scePspSaveDataDialogTerm

Terminate Save Data Dialog call for save data for PSP™

Definition

```
#include <savedata_psp_dialog.h>
SceInt32 scePspSaveDataDialogTerm()
```

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
SCE_COMMON_DIALOG_ERROR_NOT_IN_USE	0x80020411	scePspSaveDataDialogInit() is not called
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

Description

This is a version of the `sceSaveDataDialogTerm()` structure for save data for PSP™. Its basic specifications are the same as `sceSaveDataDialogTerm()`.

After calling the Save Data Dialog for save data for PSP™ with `scePspSaveDataDialogInit()`, be sure to use `scePspSaveDataDialogTerm()` to terminate the call.

Examples

```
SceCommonDialogStatus stat;

while(1) {
    stat = sceSaveDataDialogGetStatus();
    if( stat == SCE_COMMON_DIALOG_STATUS_FINISHED ) {
        scePspSaveDataDialogTerm();
        break;
    }
}
```

See Also

`scePspSaveDataDialogInit()`, `sceSaveDataDialogGetStatus()`

Constants

000004892117

Character string size

Maximum size of character strings of Save Data Dialog

Definition

Value	(Number)	Description
SCE_SAVEDATA_DIALOG_LIST_TITLE_SIZE	128	Size of character string that can be specified to <code>SceSaveDataDialogListParam.listTitle</code>
SCE_SAVEDATA_DIALOG_USER_MSG_SIZE	256	Size of character string that can be specified to <code>SceSaveDataDialogUserMessageParam.msg</code> <code>SceSaveDataDialogProgressBarParam.msg</code>

Description

Save Data Dialog 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 values above are all the size including NULL termination.

Specify all the character strings with UTF-8.

SCE CONFIDENTIAL

Save Data Slot Maximum Size

The maximum size of the save data slot

Definition

Value	(Number)	Description
SCE_SAVEDATA_DIALOG_SLOTLIST_MAXSIZE	256	The maximum size that can be specified in <code>SceSaveDataDialogListParam.slotListSize</code>

Description

This is the largest number of save data that can be handled with the list display function of Save Data Dialog.

Return Codes

List of return codes returned by Save Data Dialog

Definition

Value	(Number)	Description
SCE_COMMON_DIALOG_RESULT_OK	0x0	User selected a button. Closed with <code>sceSaveDataDialogSubClose()</code>
SCE_COMMON_DIALOG_RESULT_USER_CANCELED	0x1	User performed cancel operation
SCE_COMMON_DIALOG_RESULT_ABORTED	0x2	Aborted with <code>sceSaveDataDialogAbort()</code>
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 <code>SCE_COMMON_DIALOG_STATUS_RUNNING</code>
SCE_COMMON_DIALOG_ERROR_NOT_SUPPORTED	0x80020405	Called when the operation mode does not support the API
SCE_COMMON_DIALOG_ERROR_NOT_FINISHED	0x80020410	Called during a period other than <code>SCE_COMMON_DIALOG_STATUS_FINISHED</code>
SCE_COMMON_DIALOG_ERROR_NOT_IN_USE	0x80020411	<code>sceSaveDataDialogInit()</code> or <code>scePspSaveDataDialogInit()</code> is not called
SCE_SAVEDATA_DIALOG_ERROR_PARAM	0x80100b01	Parameter error
SCE_COMMON_DIALOG_ERROR_UNEXPECTED_FATAL	0x8002047F	Internal error

Parameter errors

Codes output when values of structures specified to Save Data Dialog are invalid

Definition

Value	Description
1	SceSaveDataDialogParam. <i>sdkVersion</i> is invalid ScePspSaveDataDialogParam. <i>sdkVersion</i> is invalid
2	SceSaveDataDialogParam. <i>mode</i> is invalid ScePspSaveDataDialogParam. <i>mode</i> is invalid
3	SceSaveDataDialogParam. <i>dispType</i> is invalid ScePspSaveDataDialogParam. <i>dispType</i> is invalid
4	SceSaveDataDialogParam. <i>flag</i> is invalid ScePspSaveDataDialogParam. <i>flag</i> is invalid
5	SceSaveDataDialogParam. <i>reserved</i> is invalid ScePspSaveDataDialogParam. <i>reserved</i> is invalid
6	SceAppUtilSaveDataSlotId is invalid SceAppUtilPspSaveDataDirName is invalid
7	SceAppUtilSaveDataSlot. <i>status</i> is invalid
8	SceAppUtilSaveDataSlotEmptyParam. <i>title</i> is invalid
9	SceAppUtilSaveDataSlotEmptyParam. <i>iconPath</i> is invalid
10	SceAppUtilSaveDataSlotEmptyParam. <i>iconBuf</i> is invalid
11	SceAppUtilSaveDataSlotEmptyParam. <i>iconBufSize</i> is invalid
12	SceAppUtilSaveDataSlotEmptyParam. <i>reserved</i> is invalid
13	SceSaveDataDialogFixedParam is NULL ScePspSaveDataDialogFixedParam is NULL
14	SceSaveDataDialogFixedParam is not NULL ScePspSaveDataDialogFixedParam is not NULL
15	SceSaveDataDialogFixedParam. <i>reserved</i> is invalid ScePspSaveDataDialogFixedParam. <i>reserved</i> is invalid
16	SceSaveDataDialogListParam is NULL ScePspSaveDataDialogListParam is NULL
17	SceSaveDataDialogListParam is not NULL ScePspSaveDataDialogListParam is not NULL
18	SceSaveDataDialogListParam. <i>slotList</i> is invalid ScePspSaveDataDialogListParam. <i>targetDirNameList</i> is invalid
19	SceSaveDataDialogListParam. <i>slotListSize</i> is invalid ScePspSaveDataDialogListParam. <i>dirNameListSize</i> is invalid
20	SceSaveDataDialogListParam. <i>focusPos</i> is invalid ScePspSaveDataDialogListParam. <i>focusPos</i> is invalid
21	SceSaveDataDialogListParam. <i>focusId</i> is invalid ScePspSaveDataDialogListParam. <i>focusDirName</i> is invalid
22	SceSaveDataDialogListParam. <i>listTitle</i> is invalid ScePspSaveDataDialogListParam. <i>listTitle</i> is invalid
23	SceSaveDataDialogListParam. <i>reserved</i> is invalid ScePspSaveDataDialogListParam. <i>reserved</i> is invalid
24	SceSaveDataDialogUserMessageParam is NULL ScePspSaveDataDialogUserMessageParam is NULL
25	SceSaveDataDialogUserMessageParam is not NULL ScePspSaveDataDialogUserMessageParam is not NULL
26	SceSaveDataDialogUserMessageParam. <i>buttonType</i> is invalid ScePspSaveDataDialogUserMessageParam. <i>buttonType</i> is invalid

Value	Description
27	SceSaveDataDialogUserMessageParam. <i>msg</i> is NULL ScePspSaveDataDialogUserMessageParam. <i>msg</i> is NULL
28	SceSaveDataDialogUserMessageParam. <i>msg</i> format is invalid ScePspSaveDataDialogUserMessageParam. <i>msg</i> format is invalid
29	SceSaveDataDialogUserMessageParam. <i>reserved</i> is invalid ScePspSaveDataDialogUserMessageParam. <i>reserved</i> is invalid
30	SceSaveDataDialogSystemMessageParam is NULL ScePspSaveDataDialogSystemMessageParam is NULL
31	SceSaveDataDialogSystemMessageParam is not NULL ScePspSaveDataDialogSystemMessageParam is not NULL
32	SceSaveDataDialogSystemMessageParam. <i>sysMsgType</i> is invalid ScePspSaveDataDialogSystemMessageParam. <i>sysMsgType</i> is invalid
33	SceSaveDataDialogSystemMessageParam. <i>value</i> is invalid ScePspSaveDataDialogSystemMessageParam. <i>value</i> is invalid
34	SceSaveDataDialogSystemMessageParam. <i>reserved</i> is invalid ScePspSaveDataDialogSystemMessageParam. <i>reserved</i> is invalid
35	SceSaveDataDialogErrorCodeParam is NULL ScePspSaveDataDialogErrorCodeParam is NULL
36	SceSaveDataDialogErrorCodeParam is not NULL ScePspSaveDataDialogErrorCodeParam is not NULL
37	SceSaveDataDialogErrorCodeParam. <i>errorCode</i> is invalid ScePspSaveDataDialogErrorCodeParam. <i>errorCode</i> is invalid
38	SceSaveDataDialogErrorCodeParam. <i>reserved</i> is invalid ScePspSaveDataDialogErrorCodeParam. <i>reserved</i> is invalid
39	SceSaveDataDialogProgressBarParam is NULL ScePspSaveDataDialogProgressBarParam is NULL
40	SceSaveDataDialogProgressBarParam is not NULL ScePspSaveDataDialogProgressBarParam is not NULL
41	SceSaveDataDialogProgressBarParam. <i>barType</i> is invalid ScePspSaveDataDialogProgressBarParam. <i>barType</i> is invalid
42	SceSaveDataDialogProgressBarParam. <i>sysMsgParam</i> is invalid ScePspSaveDataDialogProgressBarParam. <i>sysMsgParam</i> is invalid
43	SceSaveDataDialogProgressBarParam. <i>msg</i> is NULL ScePspSaveDataDialogProgressBarParam. <i>msg</i> is NULL
44	SceSaveDataDialogProgressBarParam. <i>msg</i> is not NULL ScePspSaveDataDialogProgressBarParam. <i>msg</i> is not NULL
45	SceSaveDataDialogProgressBarParam. <i>msg</i> format is invalid ScePspSaveDataDialogProgressBarParam. <i>msg</i> format is invalid
46	SceSaveDataDialogProgressBarParam. <i>reserved</i> is invalid ScePspSaveDataDialogProgressBarParam. <i>reserved</i> is invalid
51	SceSaveDataDialogFinishParam. <i>flag</i> is invalid
52	SceSaveDataDialogFinishParam. <i>reserved</i> is invalid
53	SceSaveDataDialogListParam. <i>itemStyle</i> is invalid ScePspSaveDataDialogListParam. <i>itemStyle</i> is invalid
54	SceSaveDataDialogSlotConfigParam. <i>mountPoint</i> is invalid
56	SceSaveDataDialogSlotConfigParam. <i>reserved</i> is invalid
57	SceSaveDataDialogSlotConfigParam. <i>appSubDir->srcDir</i> is invalid
58	SceSaveDataDialogSlotConfigParam. <i>appSubDir->dstDir</i> is invalid
59	SceSaveDataDialogSlotConfigParam. <i>appSubDir->reserved</i> is invalid
100	SceSaveDataDialogResult. <i>reserved</i> is invalid ScePspSaveDataDialogResult. <i>reserved</i> is invalid
101	SceSaveDataDialogResult. <i>slotInfo->reserved</i> is invalid ScePspSaveDataDialogResult. <i>paramSfo->reserved</i> is invalid

Value	Description
102	SceSaveDataDialogProgressBarTarget is invalid
103	SceSaveDataDialogParam.commonParam.bgColor is not NULL ScePspSaveDataDialogParam.commonParam.bgColor is not NULL
105	ScePspSaveDataDialogResult.paramSfo is not NULL

Description

If the contents of the structures specified for the various functions provided by Save Data Dialog are invalid, they are processed as parameter errors, and the operation status of Save Data Dialog changes immediately to SCE_COMMON_DIALOG_STATUS_FINISHED.

At this time, SCE_SAVEDATA_DIALOG_ERROR_PARAM returns to the return value of the called API or SceSaveDataDialogResult.result which can be obtained with sceSaveDataDialogGetResult() and ScePspSaveDataDialogResult.result which can be obtained with scePspSaveDataDialogGetResult().

Furthermore, the concrete parameter error occurrence locations are output to the console in the following format.

***** SceSaveDataDialog Parameter Error : XX ***** (XX is one of the above numbers)

Parameter errors are coding mistakes of the application and must absolutely be fixed before release.

Specification of Save Data

SCE CONFIDENTIAL

SceAppUtilSaveDataSlotId

Save data slot ID type

Definition

```
#include <apputil.h>
typedef SceUInt32 SceAppUtilSaveDataSlotId;
```

Description

This type specifies specific save data in Save Data Dialog in general.

The scope of the value that can be specified is $0 \leq id < \text{SCE_APPUTIL_SAVEDATA_SLOT_MAX}$. Also, `SCE_APPUTIL_SAVEDATA_SLOT_ID_INVALID` is defined as the constant indicating an invalid value.

See Also

`SceAppUtilSaveDataSlot`, `SceSaveDataDialogListParam`, `SceSaveDataDialogResult`

SCE CONFIDENTIAL

SceAppUtilSaveDataSlotEmptyParam

Empty slot specification structure

Definition

```
#include <apputil.h>
typedef struct SceAppUtilSaveDataSlotEmptyParam {
    SceChar8 *title;
    SceChar8 *iconPath;
    void *iconBuf;
    SceSize iconBufSize;
    SceChar8 reserved[32];
} SceAppUtilSaveDataSlotEmptyParam;
```

Members

<i>title</i>	Title character string displayed in empty slot (NULL termination, UTF-8)
<i>iconPath</i>	Specification of thumbnail image file
<i>iconBuf</i>	Thumbnail image buffer
<i>iconBufSize</i>	Thumbnail image buffer size
<i>reserved</i>	Reserved area (fill with all 0s)

Description

This structure specifies the empty slot for display if the specified display target save data does not exist, for the whole Save Data Dialog.

In *title*, specify the title character string to be displayed in the empty slot. If NULL is specified, the system default title is displayed. The maximum character string length (including NULL termination) that can be specified is defined in `SCE_APPUTIL_SAVEDATA_SLOT_TITLE_MAXSIZE`.

The thumbnail image file to be displayed in the empty slot is specified with a file path in *iconPath*. [a-z], [A-Z], [0-9], '-' (hyphen), '_' (underscore), '.' (dot), '/' (slash) and ':' (colon) can be used. Relative reference using "." is not possible. The longest path length that can be specified is defined in `SCE_APPUTIL_SAVEDATA_SLOT_ICON_PATH_MAXSIZE`.

In *iconBuf*, specify with the memory the thumbnail image file to be displayed in the empty slot. The image file must be placed on the memory beforehand as a PNG file. If NULL is specified, the system default thumbnail is displayed. It cannot be specified simultaneously with *iconPath*.

In *iconBufSize*, specify the size of the image file placed in *iconBuf*. `SAVEDATA_ICON_FILE_SIZE_MAX` is the maximum specifiable size. If NULL is specified in *iconBuf*, be sure to specify 0.

See Also

SceAppUtilSaveDataSlot

SCE CONFIDENTIAL

SceAppUtilSaveDataSlot

Save data item slot structure

Definition

```
#include <apputil.h>
typedef struct SceAppUtilSaveDataSlot {
    SceAppUtilSaveDataSlotId id;
    SceAppUtilSaveDataSlotStatus status;
    SceInt32 userParam;
    SceAppUtilSaveDataSlotEmptyParam *emptyParam;
} SceAppUtilSaveDataSlot;
```

Members

id Save data slot ID
status Slot status
userParam User parameter
emptyParam Empty slot specification structure

Description

This structure specifies specific save data slots in the whole Save Data Dialog.

In *id*, specify the target save data slot ID

To *status* specify the state of the save data slot. It can be specified with the following `SceAppUtilSaveDataSlotStatus` type values:

Value	(Number)	Description
<code>SCE_APPUTIL_SAVEDATA_SLOT_STATUS_AVAILABLE</code>	0	Available status
<code>SCE_APPUTIL_SAVEDATA_SLOT_STATUS_BROKEN</code>	1	Broken status

When the entity to the `SceAppUtilSaveDataSlotEmptyParam` structure is specified in *emptyParam*, and the save data slot does not exist, empty save data display is done based on the structure's setting value. For details on the display contents, refer to the `SceAppUtilSaveDataSlotEmptyParam` section.

If the save data slot exists, save data display is performed based on the information of that save data slot, so the information of *emptyParam* is ignored.

Since *userParam* is not currently used, set 0.

If NULL is specified in *emptyParam*, display of that save data is not done if the save data slot does not exist.

See Also

`SceAppUtilSaveDataSlotEmptyParam`, `SceAppUtilSaveDataSlotId`

SCE CONFIDENTIAL

SceAppUtilMountPoint

Mount point structure

Definition

```
#include <apputil.h>
typedef struct SceAppUtilMountPoint {
    SceChar8 data[SCE_APPUTIL_MOUNTPOINT_DATA_MAXSIZE];
} SceAppUtilMountPoint;
```

Members

data Mount point name

Description

This structure indicates a mount point. It is used to specify the location of the save data slot which will be displayed by Save Data Dialog by specifying `SceSaveDataDialogSlotConfigParam`. Specify either "savedata0:" or "savedata1:" for *data* as a character string.

See Also

`SceSaveDataDialogSlotConfigParam`

Specification of Save Data for PSP™

SCE CONFIDENTIAL

SceAppUtilPspSaveDataDirName

Save data directory name structure for save data for PSP™

Definition

```
#include <apputil/apputil_psp.h>
typedef struct SceAppUtilPspSaveDataDirName {
    SceChar8 data[SCE_APPUTIL_PSP_SAVEDATA_PARAMSFO_DIRECTORY_SIZE];
} SceAppUtilPspSaveDataDirName;
```

Members

data Directory name of save data for PSP™ (NULL-terminated, ASCII)

Description

This structure stores the save data directory for save data for PSP™. It is used throughout Save Data Dialog when specifying save data for PSP™.

For *data*, specify the directory name of the target save data for processing. Depending on the functionality, it is also used for specifying a save data directory name prefix.

See Also

ScePspSaveDataDialogListParam, ScePspSaveDataDialogResult,
ScePspSaveDataDialogFixedParam, ScePspSaveDataDialogUserMessageParam,
ScePspSaveDataDialogSystemMessageParam, ScePspSaveDataDialogErrorCodeParam,
ScePspSaveDataDialogProgressBarParam