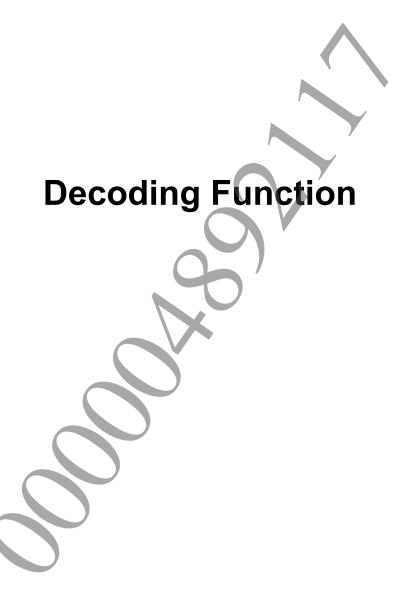


© 2011 Sony Computer Entertainment Inc. All Rights Reserved. SCE Confidential

# **Table of Contents**

Decodir	ng Function		3
	scePngDec		
Decodir	ng Auxiliary Function		6
	scePngGetOutputInfo		7
	scePngConvertToRGBA		8
Encodir	ng Auxiliary Function		9
	scePngGetEncSize		
Encodir	ng Function		11
	scePngEnc		12
Constar	nts		13
	SCE_PNG_FORMAT_BPP_MASK		14
	SCE_PNG_FORMAT_FLAG_ALPHA		15
	SCE_PNG_FORMAT_FLAG_CLUT		16
	SCE_PNG_FORMAT_FLAG_GRAYSCALE		
	SCE_PNG_FORMAT_FLAG_INTERLACE	<u> </u>	18
	Return Codes		19



# scePngDec

# PNG decoding function

#### **Definition**

# **Arguments**

dst	Pointer to buffer for storing decoding results
outsize	Size of buffer for storing decoding results
src	Pointer to buffer where PNG data is stored
insize	Size of PNG data
width	Pointer to destination where width (in pixels) of decoding result data is stored
height	Pointer to destination where height (in pixels) of decoding result data is stored
format	Pointer to destination where format of decoding result data is stored

#### **Return Value**

When processing completes successfully, the number of bytes of decoding result data is returned. A negative number indicates one of the following errors.

Value	Hexadecimal	Description
SCE_PNG_ERROR_INVALID_FORMAT	0x80690001	Could not perform decoding because data
		was invalid
SCE_PNG_ERROR_INVALID_SIZE	0x80690002	Invalid buffer size was specified
SCE_PNG_ERROR_NOT_SUPPORTED	0x80690003	Could not perform decoding because PNG
		data contains a format that is not supported
SCE_PNG_ERROR_INVALID_POINTER	0x80690004	Invalid buffer pointer

# **Description**

This function expands the given PNG data and converts it to raster scan format (or ADAM7 interlace scan format). If the input PNG data also uses a CLUT (PALETTE), the contents of the CLUT are also included in the output.

The output data begins with the CLUT data (if present) followed by the image data. For details, refer to the "SCEPNG Overview" document.

The data format of the decoded result is stored in \*format. This will be the logical OR of the following values.

```
Number of bits per pixel (1 to 64)
```

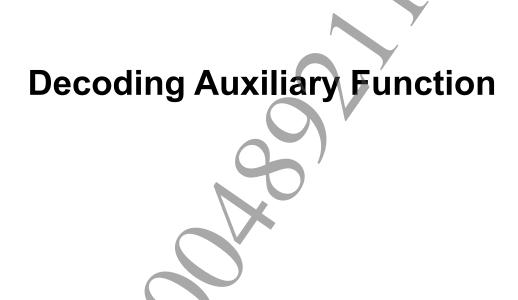
```
SCE_PNG_FORMAT_FLAG_ALPHA
SCE_PNG_FORMAT_FLAG_CLUT
SCE_PNG_FORMAT_FLAG_GRAYSCALE
SCE_PNG_FORMAT_FLAG_INTERLACE
```

**©SCEI** 

# Notes

This function is multithread safe.





# scePngGetOutputInfo

Obtain PNG Decoder output information

#### **Definition**

# **Arguments**

src Pointer to buffer where PNG data is stored

insize PNG data size

width Pointer to location where width (number of pixels) of decoded result data is stored height Pointer to location where height (number of pixels) of decoded result data is stored

outputFormat Pointer to location where format of decoded result data is stored

streamFormat Pointer to location where stream format is stored

#### **Return Values**

During normal behavior, this function returns the number of bytes required for storing the decoded result data.

For negative values, one of the following errors will be returned.

Value	Hexadecimal	Description
SCE_PNG_ERROR_INVALID_FORMAT	0x80690001	Cannot perform decoding because data is
		invalid
SCE_PNG_ERROR_INVALID_SIZE	0x80690002	Invalid buffer size was specified
SCE_PNG_ERROR_NOT_SUPPORTED	0x80690003	Information cannot be obtained because the
		PNG format of the data is unsupported
SCE_PNG_ERROR_INVALID_POINTER	0x80690004	Invalid buffer pointer

# **Description**

After reading the header information of the PNG data provided, obtain the image size, the data format and the number of bytes required for the storage buffer for the decoded result data.

The data format of the decoded result data is stored in \*outputFormat. This value is equivalent to the value of \*format obtained by scePngDec(). Refer to the description of \*format in scePngDec() regarding this value.

The stream data format is stored in \*streamFormat. If the stream data format is RGB888, this value will differ from the value of \*outputFormat because the data format is converted to RGBA8888 during decoding. Refer to the description of \*format in scePngDec() regarding this value.

## **Notes**

This function is multithread safe.

**©SCEI** 

# scePngConvertToRGBA

Convert CLUT/grayscale/ADAM7 interlace to RGBA8888 format.

#### **Definition**

# **Arguments**

dst Pointer to buffer for storing converting results

src Pointer to buffer for storing decoding result data

width Width (in pixels) of decoding result data

height Height (in pixels) of decoding result data

format Format of decoding result data

#### **Return Values**

When processing completes successfully, 0 is returned. A negative number indicates one of the following errors.

Value	Hexadecimal	Description
SCE_PNG_ERROR_INVALID_FORMAT	0x80690001	Cannot perform decoding because data is
		invalid
SCE_PNG_ERROR_NOT_SUPPORTED	0x80690003	Could not perform converting because
		format is not supported
SCE_PNG_ERROR_INVALID_POINTER	0x80690004	Invalid buffer pointer

# **Description**

This function converts decode results obtained from scePngDec() to RGBA8888 format.

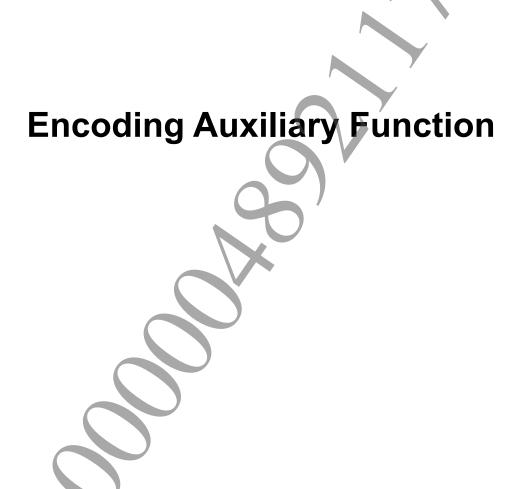
If the given data is in CLUT (PALETTE) or grayscale format, it is converted to RGBA8888. If the given data is in ADAM7 interlace scan format, it is rearranged in raster scan format.

To src, width, height and format, specify the decode results obtained from scePngDec().

To dst, specify where the converted results will be stored. The buffer size will not be checked. Specify the buffer where the specified image size using width and height in RGBA8888 format can be stored.

# **Notes**

This function is multithread safe.



# Document serial number: 000004892117

# scePngGetEncSize

Estimate size of PNG encoding result

#### **Definition**

# **Arguments**

width Width (in pixels) of data to be encoded height Height (in pixels) of data to be encoded

#### **Return Value**

When processing completes successfully, the maximum data size (in bytes) of the result of encoding an image of the given size is returned.

A negative number indicates the following error.

Value	Hexadecimal Description	
SCE_PNG_ERROR_INVALID_SIZE	0x80690002 Invalid image size was specifi	ed

# **Description**

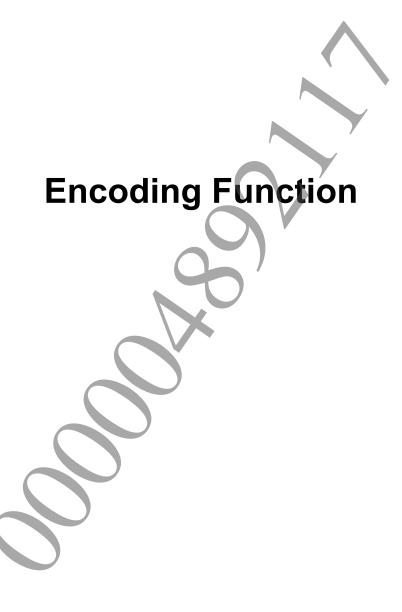
This function returns an estimate of the size of the encoding result. Normally, the size of data generated by scePngEnc() will not exceed the return value of this function (however, it may be several bytes smaller).

By calling this function before performing encoding, you can check whether the output buffer size will be large enough.

#### **Notes**

This function is multithread safe





# scePngEnc

# PNG encoding function

## **Definition**

# **Arguments**

dst Pointer to buffer for storing encoding result

src Pointer to frame buffer which contains the image to be encoded

width
 height
 framewidth
 format
 Width (in pixels) of data to be encoded
 Width (in pixels) of frame buffer
 Pixel format of data to be encoded

#### **Return Value**

When processing completes successfully, the number of bytes of encoding result data is returned. A negative number indicates the following error.

Value	Hexadecimal	Description
SCE_PNG_ERROR_INVALID_SIZE	0x80690002	Invalid image size was specified
SCE_PNG_ERROR_NOT_SUPPORTED	0x80690003	Could not perform encoding because format is not supported by this library
SCE_PNG_ERROR_INVALID_POINTER	0x80690004	Invalid buffer pointer

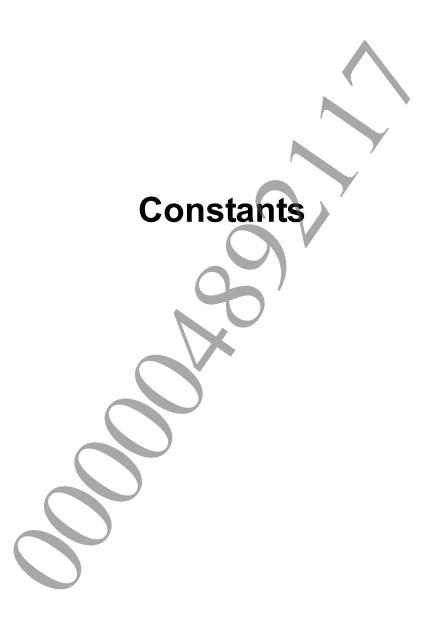
## Description

This function converts input data in raster scan format (RGBA888) to uncompressed PNG format (RGB888).

## **Notes**

This function is multithread safe.

The alpha information included in the input data is constantly discarded.



# SCE\_PNG\_FORMAT\_BPP\_MASK

Value used to get number of bits from return value indicated by format

## **Definition**

```
#include <scepng.h>
#define SCE_PNG_FORMAT_BPP_MASK (0xff)
```

# **Description**

The number of bits per pixel in the image data that is output can be obtained by taking the logical AND of this value and the value of format returned from scePngDec().

## See Also



# SCE\_PNG\_FORMAT\_FLAG\_ALPHA

Flag indicating whether alpha information is included

## **Definition**

```
#include <scepng.h>
#define SCE_PNG_FORMAT_FLAG_ALPHA (0x400)
```

# **Description**

Whether alpha information is included in the image data that is output can be known by taking the logical AND of this value and the value of format returned from scePngDec().

## See Also



# SCE\_PNG\_FORMAT\_FLAG\_CLUT

Flag indicating whether CLUT is included

## **Definition**

```
#include <scepng.h>
#define SCE_PNG_FORMAT_FLAG_CLUT (0x100)
```

# **Description**

Whether a CLUT is included in the image data that is output can be known by taking the logical AND of this value and the value of format returned from scePngDec().

## See Also



# SCE\_PNG\_FORMAT\_FLAG\_GRAYSCALE

Flag indicating whether image is in grayscale format

## **Definition**

#include <scepng.h>
#define SCE\_PNG\_FORMAT\_FLAG\_GRAYSCALE (0x200)

# **Description**

Whether the image data that is output is in grayscale format can be known by taking the logical AND of this value and the value of format returned from scePngDec().

## See Also



# SCE\_PNG\_FORMAT\_FLAG\_INTERLACE

Flag indicating whether image is interlaced

## **Definition**

#include <scepng.h>
#define SCE\_PNG\_FORMAT\_FLAG\_INTERLACE (0x1000)

# **Description**

Whether the image data that is output is interlaced can be known by taking the logical AND of this value and the value of format returned from scePngDec().

## See Also



# **Return Codes**

List of return codes returned by SCEPNG

# Definition

Value	Hexadecimal	Description
SCE_PNG_ERROR_INVALID_FORMAT	0x80690001	Invalid data format
SCE_PNG_ERROR_INVALID_SIZE	0x80690002	An invalid size was specified
SCE_PNG_ERROR_NOT_SUPPORTED	0x80690003	Unsupported data format
SCE PNG ERROR INVALID POINTER	0x80690004	Invalid buffer pointer

