libscebase Reference

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Table of Contents

Primitive Types

SceUChar8, SceByte8, SceUShort16, SceUInt16, SceUInt32, SceUInt64, SceULong64

Unsigned integer

Definition

```
#include <scebase.h>
typedef SceUChar8;
typedef SceByte8;
typedef SceUShort16;
typedef SceUInt16;
typedef SceUInt32;
typedef SceUInt64;
typedef SceULong64;
```

Description

SceUChar8 and SceByte8 represent 8-bit unsigned integers.

SceUShort16 and SceUInt16 represent 16-bit unsigned integers.

SceUInt32 represents 16-bit unsigned integers.

SceUInt64 and SceULong64 represent 64-bit unsigned integers.

See Also

SceChar8, SceSByte8, SceShort16, SceInt16, SceInt32, SceInt64, SceLong64

SceChar8, SceSByte8, SceShort16, SceInt16, SceInt32, SceInt64, SceLong64

Signed integer

Definition

```
#include <scebase.h>
typedef SceChar8;
typedef SceSByte8;
typedef SceShort16;
typedef SceInt16;
typedef SceInt32;
typedef SceInt64;
typedef SceLong64;
```

Description

SceChar8 and SceSByte8 represent 8-bit signed integers.

SceShort16 and SceInt16 represent 16-bit signed integers.

SceInt32 represents 32-bit signed integers.

SceInt64 and SceLong64 represent 64-bit signed integers.

See Also

SceUChar8, SceByte8, SceUShort16, SceUInt16, SceUInt32, SceUInt64, SceULong64

SceFloat, SceFloat32, SceDouble, SceDouble64

Floating-point number

Definition

```
#include <scebase.h>
typedef SceFloat;
typedef SceFloat32;
typedef SceDouble;
typedef SceDouble64;
```

Description

SceFloat and SceFloat32 represent 32-bit floating-point numbers.

SceDouble and SceDouble64 represent 64-bit floating-point numbers.

SceBool

Boolean value

Definition

```
#include <scebase.h>
typedef int SceBool;
```

Description

This type represents a Boolean value. The following macros are defined in scebase.h.

```
#define SCE_FALSE   0
#define SCE_TRUE    1
```

A variable of type SceBool must have one of these two values.

Also, the value must not depend on sizeof(SceBool).

SceVoid, ScePVoid

void and pointer to void

Definition

```
#include <scebase.h>
typedef void SceVoid;
typedef void *ScePVoid;
```

Description

This type is a synonym for type `void`.

This type is a synonym for type `(void *)`.

SceWChar16, SceWChar32

Wide character

Definition

```
#include <scebase.h>
typedef SceWChar16;
typedef SceWChar32;
```

Description

`SceWChar16` represents 16-bit wide characters.

`SceWChar32` represents 32-bit wide characters.

Vector Type Structures

SceIVector2, SceIVector3, SceIVector4

Vector of 32-bit integers

Definition

```
#include <scebase.h>
typedef struct SceIVector2 {
        int x, y;
} SceIVector2;

typedef struct SceIVector3 {
        int x, y, z;
} SceIVector3;

typedef struct SceIVector4 {
        int x, y, z, w;
} SceIVector4;
```

Description

SceIVector2 represents 2D vectors of 32-bit integers.

SceIVector3 represents 3D vectors of 32-bit integers.

SceIVector4 represents 4D vectors of 32-bit integers.

See Also

SceUVector2, SceUVector3, SceUVector4

SceFVector2, SceFVector3, SceFVector4

Vector of 32-bit floating-point numbers

Definition

```
#include <scebase.h>
typedef struct SceFVector2 {
        float x, y;
} SceFVector2;

typedef struct SceFVector3 {
        float x, y, z;
} SceFVector3;

typedef struct SceFVector4 {
        float x, y, z, w;
} SceFVector4;
```

Description

`SceFVector2` represents 2D vectors of 32-bit floating-point numbers.

`SceFVector3` represents 3D vectors of 32-bit floating-point numbers.

`SceFVector4` represents 4D vectors of 32-bit floating-point numbers.

See Also

`SceUVector2`,`SceUVector3`,`SceUVector4`

Matrix Type Structures

SceIMatrix2, SceIMatrix3, SceIMatrix4

Matrix of 32-bit integers

Definition

```
#include <scebase.h>
typedef struct SceIMatrix2 {
        SceIVector2 x, y;
} SceIMatrix2;

typedef struct SceIMatrix3 {
        SceIVector3 x, y, z;
} SceIMatrix3;

typedef struct SceIMatrix4 {
        SceIVector4 x, y, z, w;
} SceIMatrix4;
```

Description

SceIMatrix2 represents 2x2 matrices of 32-bit integers.

SceIMatrix3 represents 3x3 matrices of 32-bit integers.

SceIMatrix4 represents 4x4 matrices of 32-bit integers.

See Also

SceIVector2, SceUMatrix2, SceIVector3, SceUMatrix3, SceIVector4, SceUMatrix4

SceFMatrix2, SceFMatrix3, SceFMatrix4

Matrix of 32-bit floating-point numbers

Definition

```
#include <scebase.h>
typedef struct SceFMatrix2 {
        SceFVector2 x, y;
} SceFMatrix2;

typedef struct SceFMatrix3 {
        SceFVector3 x, y, z;
} SceFMatrix3;

typedef struct SceFMatrix4 {
        SceFVector4 x, y, z, w;
} SceFMatrix4;
```

Description

`SceFMatrix2` represents 2x2 matrices of 32-bit floating-point numbers.

`SceFMatrix3` represents 3x3 matrices of 32-bit floating-point numbers.

`SceFMatrix4` represents 4x4 matrices of 32-bit floating-point numbers.

See Also

`SceFVector2`, `SceUMatrix2`, `SceFVector3`, `SceUMatrix3`, `SceFVector4`, `SceUMatrix4`

Quaternion Type Structures

SceFQuaternion

Quaternion vector of 32-bit floating-point number

Definition

```
#include <scebase.h>
typedef struct SceFQuaternion {
        float x, y, z, w;
} SceFQuaternion;
```

Description

This type represents quaternion of floating-point numbers.

x, *y*, *z* represent the axis of rotation, and *w* represents the angle of rotation.

The relation between a unit quaternion q and a rotation, which is represented by a normalized rotation axis vector and angle θ, is given by the following equations.

```
q.x = sin(θ/2) * axis.x
q.y = sin(θ/2) * axis.y
q.z = sin(θ/2) * axis.z
q.w = cos(θ/2)
```

Unions

SceUnion32

32-bit union

Definition

```
#include <scebase.h>
typedef union SceUnion32 {
        unsigned int ui;
        int i;
        unsigned short us[2];
        short s[2];
        unsigned char uc[4];
        char c[4];
        float f;
        void *p;
} SceUnion32;
```

Description

This is a union declaration for type converting 32-bit types.

SceUnion64

64-bit union

Definition

```
#include <scebase.h>
typedef union SceUnion64 {
        SceULong64 ull;
        SceLong64 ll;
        unsigned int ui[2];
        int i[2];
        unsigned short us[4];
        short s[4];
        unsigned char uc[8];
        char c[8];
        float f[2];

#if defined(__ARM_NEON__)
#if defined(__SNC__) || (defined(__GNUC__) && defined(_GCC_ARM_NEON_H))
        int8x8_t i8x8;
        int16x4_t i16x4;
        int32x2_t i32x2;
        int64x1_t i64x1;
        float32x2_t f32x2;
        poly8x8_t p8x8;
        poly16x4_t p16x4;
        uint8x8_t ui8x8;
        uint16x4_t ui16x4;
        uint32x2_t ui32x2;
        uint64x1_t ui64x1;
#endif  /* defined(__SNC__) || (defined(__GNUC__) && defined(_GCC_ARM_NEON_H)) */
#endif  /* defined(__ARM_NEON__) */
} SceUnion64;
```

Description

This is a union declaration for type converting 64-bit types.

SceUnion128

128-bit union

Definition

```
#include <scebase.h>
typedef union SceUnion128 {
        SceULong64 ull[2];
        SceLong64 ll[2];
        unsigned int ui[4];
        int i[4];
        unsigned short us[8];
        short s[8];
        unsigned char uc[16];
        char c[16];
        float f[4];

        SceFVector4 fv;
        SceFQuaternion fq;
        SceIVector4 iv;
        SceFPlane fp;
        SceFColor fc;

#if defined(__ARM_NEON__)
#if defined(__SNC__) || (defined(__GNUC__) && defined(_GCC_ARM_NEON_H))
        int8x16_t i8x16;
        int16x8_t i16x8;
        int32x4_t i32x4;
        int64x2_t i64x2;
        float32x4_t f32x4;
        poly8x16_t p8x16;
        poly16x8_t p16x8;
        uint8x16_t ui8x16;
        uint16x8_t ui16x8;
        uint32x4_t ui32x4;
        uint64x2_t ui64x2;
#endif  /* defined(__SNC__) || (defined(__GNUC__) && defined(_GCC_ARM_NEON_H)) */
#endif  /* defined(__ARM_NEON__) */
} SceUnion128;
```

Description

This is a union declaration for type converting 128-bit types.

SceUnion512

512-bit union

Definition

```
#include <scebase.h>
typedef union SceUnion512 {
        SceULong64 ull[8];
        SceLong64 ll[8];
        unsigned int ui[16];
        int i[16];
        unsigned short us[32];
        short s[32];
        unsigned char uc[64];
        char c[64];
        float f[16];

        SceFMatrix4 fm;
        SceIMatrix4 im;
        SceUMatrix4 um;

#if defined(__ARM_NEON__)
#if defined(__SNC__) || (defined(__GNUC__) && defined(_GCC_ARM_NEON_H))
        int32x4x4_t i32x4x4;
        float32x4x4_t f32x4x4;
#endif  /* defined(__SNC__)||(defined(__GNUC__)&&defined(_GCC_ARM_NEON_H))*/
#endif  /* defined(__ARM_NEON__) */
} SceUnion512;
```

Description

This is a union declaration for type converting 512-bit types.

SceUVector2

2D vector

Definition

```
#include <scebase.h>
typedef union SceUVector2 {
        SceIVector2 iv;
        SceFVector2 fv;
#if defined(__ARM_NEON__)
#if defined(__SNC__) || (defined(__GNUC__) && defined(_GCC_ARM_NEON_H))
        int32x2_t i32x2;
        float32x2_t f32x2;
        uint32x2_t ui32x2;
#endif  /* defined(__SNC__)||(defined(__GNUC__)&&defined(_GCC_ARM_NEON_H))*/
#endif  /* defined(__ARM_NEON__) */
} SceUVector2;
```

Description

This is a union declaration for type converting 32-bit 2D-vector types.

See Also

`SceIVector2`, `SceFVector2`

SceUVector3

3D vector

Definition

```
#include <scebase.h>
typedef union SceUVector3 {
        SceIVector3 iv;
        SceFVector3 fv;
} SceUVector3;
```

Description

This is a union declaration for type converting 32-bit 3D-vector types.

See Also

`SceIVector3`, `SceFVector3`

SceUVector4

4D vector

Definition

```
#include <scebase.h>
typedef union SceUVector4 {
        SceIVector4 iv;
        SceFVector4 fv;
#if defined(__ARM_NEON__)
#if defined(__SNC__) || (defined(__GNUC__) && defined(_GCC_ARM_NEON_H))
        int32x4_t i32x4;
        float32x4_t f32x4;
        uint32x4_t ui32x4;
#endif  /* defined(__SNC__)||(defined(__GNUC__)&&defined(_GCC_ARM_NEON_H))*/
#endif  /* defined(__ARM_NEON__) */
} SceUVector4;
```

Description

This is a union declaration for type converting 32-bit 4D-vector types.

See Also

`SceIVector4,SceFVector4`

SceUMatrix2

2D matrix

Definition

```
#include <scebase.h>
typedef union SceUMatrix2 {
        SceIMatrix2 im;
        SceFMatrix2 fm;
        SceFVector2 fv[2];
        SceIVector2 iv[2];
        SceUVector2 uv[2];
        float f[2][2];
        int i[2][2];

#if defined(__ARM_NEON__)
#if defined(__SNC__) || (defined(__GNUC__) && defined(_GCC_ARM_NEON_H))
        int32x2x2_t i32x2x2;
        float32x2x2_t f32x2x2;
        uint32x2x2_t ui32x2x2;
#endif  /* defined(__SNC__)||(defined(__GNUC__)&&defined(_GCC_ARM_NEON_H))*/
#endif  /* defined(__ARM_NEON__) */

#if defined(SCE_TYPES_USE_UNNAMED_UNION)
        struct {
              float f00, f10;
              float f01, f11;
        };
        struct {
              int i00, i10;
              int i01, i11;
        };
#endif  /* defined(SCE_TYPES_USE_UNNAMED_UNION) */
} SceUMatrix2;
```

Description

This is a union declaration for type converting 32-bit 2x2 matrix types.

See Also

`SceIVector2, SceFVector2, SceIMatrix2, SceFMatrix2`

SceUMatrix3

3D matrix

Definition

```
#include <scebase.h>
typedef union SceUMatrix3 {
        SceFMatrix3 fm;
        SceIMatrix3 im;
        SceFVector3 fv[3];
        SceIVector3 iv[3];
        SceUVector3 uv[3];
        float f[3][3];
        int i[3][3];

#if defined(SCE_TYPES_USE_UNNAMED_UNION)
        struct {
              float f00, f10, f20;
              float f01, f11, f21;
              float f02, f12, f22;
        };
        struct {
              int i00, i10, i20;
              int i01, i11, i21;
              int i02, i12, i22;
        };
#endif  /* defined(SCE_TYPES_USE_UNNAMED_UNION) */
} SceUMatrix3;
```

Description

This is a union declaration for type converting 32-bit 3x3 matrix types.

See Also

`SceIVector3,SceFVector3,SceIMatrix3,SceFMatrix3`

SceUMatrix4

4D matrix

Definition

```
#include <scebase.h>
typedef union SceUMatrix4 {
        SceFMatrix4 fm;
        SceIMatrix4 im;
        SceFVector4 fv[4];
        SceIVector4 iv[4];
        SceUVector4 uv[4];
        float f[4][4];
        int i[4][4];

#if defined(__ARM_NEON__)
#if defined(__SNC__) || (defined(__GNUC__) && defined(_GCC_ARM_NEON_H))
        int32x4x4_t i32x4x4;
        float32x4x4_t f32x4x4;
        uint32x4x4_t ui32x4x4;
#endif  /* defined(__SNC__)||(defined(__GNUC__)&&defined(_GCC_ARM_NEON_H))*/
#endif  /* defined(__ARM_NEON__) */

#if defined(SCE_TYPES_USE_UNNAMED_UNION)
        struct {
              float f00, f10, f20, f30;
              float f01, f11, f21, f31;
              float f02, f12, f22, f32;
              float f03, f13, f23, f33;
        };
        struct {
              int i00, i10, i20, i30;
              int i01, i11, i21, i31;
              int i02, i12, i22, i32;
              int i03, i13, i23, i33;
        };
#endif  /* defined(SCE_TYPES_USE_UNNAMED_UNION) */
} SceUMatrix4;
```

Description

This is a union declaration for type converting 32-bit 4x4 matrix types.

See Also

`SceIVector4,SceFVector4,SceIMatrix4,SceFMatrix4`

Time Structure

SceDateTime

Time information

Definition

```
#include <scebase.h>
typedef struct SceDateTime {
        unsigned short year;
        unsigned short month;
        unsigned short day;
        unsigned short hour;
        unsigned short minute;
        unsigned short second;
        unsigned int microsecond;
} SceDateTime;
```

Members

year	Year (1 to 9999)
month	Month (1 to 12)
day	Day (1 to 31)
hour	Hour (0 to 23)
minute	Minutes (0 to 59)
second	Seconds (0 to 59)
microsecond	Microseconds (0 to 999999)

Description

This structure is used for handling time information in a consistent manner. It is used by various libraries for converting time information.

System Types

SceUID

Unique identifier indicating kernel resource

Definition

```
#include <scebase.h>
typedef int SceUID;
```

Description

This is a `signed int` type but only a positive number is valid. Negative values are used for error codes. This can be used together with `SceError`.

SceName

Type indicating kernel resource name

Definition

```
#include <scebase.h>
typedef char *SceName;
```

Description

This is a type indicating the kernel resource name.

SceSize

Unsigned size type

Definition

```
#include <scebase.h>
typedef unsigned int SceSize;
```

Description

This is an unsigned size type.

SceSSize

Signed size type

Definition

```
#include <scebase.h>
typedef int SceSSize;
```

Description

This is a signed size type.

ScePSize

Physical address size type

Definition

```
#include <scebase.h>
typedef SceSize ScePSize;
```

Description

This is a physical address size type.

ScePSSize

Signed physical address size type

Definition

```
#include <scebase.h>
typedef SceSSize **ScePSSize**;
```

Description

This is a signed physical address size type.

SceVSize

Virtual address size type

Definition

```
#include <scebase.h>
typedef SceSize SceVSize;
```

Description

This is a virtual address size type.

SceVSSize

Signed virtual address size type

Definition

```
#include <scebase.h>
typedef SceSSize SceVSSize;
```

Description

This is a signed virtual address size type.

SceUIntVAddr

Unsigned integer type representing virtual address

Definition

```
#include <scebase.h>
typedef unsigned int SceUIntVAddr;
```

Description

This is an unsigned integer type representing a virtual address.

SceOff

Offset type

Definition

```
#include <scebase.h>
typedef SceInt64 SceOff;
```

Description

This is an offset type.

Macros

Macros Used in the Entire SDK System

Macros defined in scebase.h or scebase_common.h

Definition

Value	Hexadecimal	Description

`SCE_OK`	0x0	Value returned when a function is processed successfully
		#include <scebase.h>
		or
		#include <scebase_common.h>

`SCE_PSP2_SDK_VERSION`	Value of version	SDK version #include <scebase.h>