# liblocation Reference



PlayStation®Vita Programmer Tool Runtime Library Release 3.50





Document serial number: 000004892117

# Table of Contents

# Datatypes

# SceLocationHandle

Handle type

### Definition

```
#include <liblocation.h>
typedef SceUInt32 SceLocationHandle;
```

### Description

This is the handle type consisting of a number for managing liblocation processing units. The handle is obtained with `sceLocationOpen()` and is used when executing functions.
`SCE_LOCATION_HANDLE_INVALID (0xFFFFFFFF)` is an invalid value, and will return if calling `sceLocationOpen()` results in an error. When liblocation is executed with several different methods, execute `sceLocationOpen()` for the required number of times, and sort the processing according to each of the obtained handles.

# SceLocationLocationInfo

Location information

### Definition

```
#include <liblocation.h>
struct {
        SceDouble64 latitude,
        SceDouble64 longitude,
        SceDouble64 altitude,
        SceFloat32 accuracy,
        SceFloat32 reserve,
        SceFloat32 direction,
        SceFloat32 speed,
        SceRtcTick timestamp
} SceLocationLocationInfo;
```

### Members

| | |
|---|---|
| *latitude* | Latitude (deg). Valid range: -90 to + 90<br>If not obtained, SCE_LOCATION_DATA_INVALID is stored |
| *longitude* | Longitude (deg). Valid range: -180 to +180<br>If not obtained, SCE_LOCATION_DATA_INVALID is stored |
| *altitude* | Altitude (m)<br>If not obtained, SCE_LOCATION_DATA_INVALID is stored |
| *accuracy* | Horizontal error (m).<br>If not obtained, SCE_LOCATION_DATA_INVALID is stored |
| *reserve* | Reserved area |
| *direction* | Travel direction. Clockwise angle (0 to 360 degrees) from north.<br>If not obtained, SCE_LOCATION_DATA_INVALID is stored |
| *speed* | Travel speed (km/h).<br>If not obtained, SCE_LOCATION_DATA_INVALID is stored |
| *timestamp* | Time of data obtainment in usec (UTC) |

### Description

This is a structure that represents location information.

Even when values can be obtained for other member values, *direction* and *speed* may be SCE_LOCATION_DATA_INVALID.

# SceLocationLocationInfoCallback

Location information callback notification function

## Definition

```
#include <liblocation.h>
void (*SceLocationLocationInfoCallback)(
        SceInt32 result,
        SceLocationHandle handle,
        const SceLocationLocationInfo *location,
        void *userdata
)
```

## Arguments

| | |
|---|---|
| *result* | Result of location information obtainment processing |
| *handle* | Handle |
| *location* | Obtained location information |
| *userdata* | userdata specified at start |

## Return Values

The following values are stored to the *result* argument.

Returns SCE_LOCATION_SUCCESS for success.

Returns mainly the following values in case of errors.

| Value | Description |
|---|---|
| SCE_LOCATION_INFO_UNDETERMINED_LOCATION | Could not get information for location calculation |
| SCE_LOCATION_INFO_DENIED_BY_USER | Location information usage is not allowed by the user |
| SCE_LOCATION_INFO_DISABLE_DEVICE | Location calculating device is disabled and location cannot be calculated |

## Description

This is a callback function that notifies location information during continual location obtainment.

Normally, a callback occurs only once for an error and will next occur when a value is obtained or another error occurs. However, when the return value is SCE_LOCATION_INFO_DENIED_BY_USER, a callback also occurs when the cause of the error changes. For example, if the use of location information is restricted by both the Parental Controls application and Settings application, the restriction will remain in place due to the setting in the Settings application even if the restriction is removed in the Parental Controls application, and a callback of SCE_LOCATION_INFO_DENIED_BY_USER will occur again.

The calculation result given here is a calculated status output during a point in continual calculation. For example, even when SCE_LOCATION_INFO_UNDETERMINED_LOCATION returns immediately after beginning calculation, this may just mean that calculation is still in progress; thus, it is not recommended to conclude with this result that location calculation is not possible.

If SCE_LOCATION_MODE_CONFIRM_AUTO is specified in sceLocationInit()'s *mode*, the dialog for obtaining permission to use location information may be displayed before callback is notified.

# SceLocationHeadingInfo

Direction information

### Definition

```
#include <liblocation.h>
struct {
        SceFloat32 heading,
        SceFQuaternion nedQuat,
        SceUInt32 stability,
        SceRtcTick timestamp
} SceLocationHeadingInfo;
```

### Members

| | |
|---|---|
| *heading* | Clockwise angle (0 to 360 degrees) from north indicated by the axis of the local device.<br>If not obtained, SCE_LOCATION_DATA_INVALID is stored |
| *nedQuat* | Quaternion indicating the direction indicated by the axis of the local device<br>If not obtained, SCE_LOCATION_DATA_INVALID is stored in each element |
| *stability* | Calibration index value for magnetometer |
| *timestamp* | Time of data obtainment in μsec (UTC) |

### Description

This structure represents direction information. *heading* and *nedQuat* indicate the direction indicated by the axis of the local device, but the axis of the local device varies according to SceLocationHeadingMethod of *hmethod* specified with sceLocationOpen(). In the case of SCE_LOCATION_HMETHOD_AUTO or SCE_LOCATION_HMETHOD_VERTICAL, SCE_LOCATION_HMETHOD_HORIZONTAL, the direction orthogonal to the longitudinal direction in the horizontal plane of the device is used as the axis. In the case of SCE_LOCATION_HMETHOD_CAMERA, the direction indicated by the camera that is orthogonal to the horizontal plane of the device is used as the axis. When SCE_LOCATION_HEADING_NORTH_TRUE is set in *hmethod*, declination according to the measuring site is added to the north direction, which is used as the reference, to obtain the true north. *stability* changes when the stability of the value of *heading* or *nedQuat* changes.

# SceLocationHeadingInfoCallback

Callback notification function for direction information

### Definition

```
#include <liblocation.h>
void (*SceLocationHeadingInfoCallback)(
        SceInt32 result,
        SceLocationHandle handle,
        const SceLocationHeadingInfo *heading,
        void *userdata
)
```

### Arguments

| | |
|---|---|
| *result* | Result of direction information obtainment processing |
| *handle* | Handle |
| *heading* | Obtained direction information |
| *userdata* | userdata specified at start |

### Return Values

The following values are stored to the *result* argument.

Returns SCE_LOCATION_SUCCESS for success.

Returns mainly the following values in case of errors.

| Value | Description |
|---|---|
| SCE_LOCATION_ERROR_INVALID_ADDRESS | Invalid data storing address |
| SCE_LOCATION_INFO_UNDETERMINED_HEADING | Could not get direction information |

### Description

This is a callback function that notifies the direction information during continual direction obtainment.

# SceLocationPermissionInfo

Location information usage permission information

## Definition

```
#include <liblocation.h>
struct {
        SceLocationPermissionStatus parentalstatus,
        SceLocationPermissionStatus mainstatus,
        SceLocationPermissionApplicationStatus applicationstatus,
        SceUInt32 updated,
        SceUInt32 reserve
} SceLocationPermissionInfo;
```

## Members

| | |
|---|---|
| *parentalstatus* | Status of usage permission through parental control |
| *mainstatus* | Status of usage permission through location data item of system settings |
| *applicationstatus* | Status of usage permission through location data item for each application in system settings |
| *updated* | SCE_LOCATION_PERMISSION_NOT_UPDATED is stored if the location information usage permission information has not been changed even once since the execution of sceLocationConfirm(), and SCE_LOCATION_PERMISSION_UPDATED is stored if the location information usage permission information has been changed through the system settings, etc. |
| *reserve* | Reserved area |

## Description

This structure represents the location information usage permission for each cause. For details on the members indicating the various statuses, refer to the locations describing their respective enum.

*updated* is information that can be used to prevent the dialog for obtaining permission to use location information from being displayed needlessly. Normally, if the user has selected "denied" when the dialog is displayed, it is not recommended to display the dialog again. However, if the usage permission status has been changed along the way through the **Location Data** item in **Settings** of the system software, it is desirable to enable display of the dialog for obtaining permission to use location information again. *updated* indicates whether the usage permission status has been changed through **Settings** of the system software, and in the case of the above-described status, it can be used to select whether to display again the dialog for obtaining permission to use location information. This status can be continually maintained even in the interval between termination and restart of the application or the system.

# Functions

# sceLocationInit

Initialize library

## Definition

```
#include <liblocation.h>
SceInt32 sceLocationInit (
        SceUInt32 open,
        SceUInt32 mode
)
```

## Arguments

**In**

*open* Maximum number of open handles opened by the application (enter a value of 8 or lower)
*mode* Specifies the liblocation mode

## Return Values

Returns SCE_LOCATION_SUCCESS for success.

Returns mainly the following values in case of errors.

| Value | Description |
|---|---|
| SCE_LOCATION_ERROR_INVALID_VALUE | An invalid value was entered |
| SCE_LOCATION_ERROR_NO_MEMORY | Failed to allocate memory |
| SCE_LOCATION_ERROR_INITIALIZED | liblocation has been initialized |

## Description

This function initializes liblocation. After liblocation is loaded, execute this function only once before the first execution of sceLocationOpen(). Specify the maximum number of open handles opened by the application to the argument *open*.

Since liblocation allows up to 8 open handles per application, enter a value of 8 or lower in *open*.

For *mode*, specify a macro defined by the constants SceLocationMode and SceLocationModeConfirm with OR operation.

liblocation operates by dynamically and internally allocating memory when sceLocationInit() is executed. The memory capacity at that time is the maximum number of open handles x 8 KiB + 4 KiB.

sceLocationSetThreadParameter() can be executed before this function is executed.

# sceLocationTerm

Terminate library

### Definition

```
#include <liblocation.h>
SceInt32 sceLocationTerm (
        void
)
```

### Arguments

None

### Return Values

Returns SCE_LOCATION_SUCCESS for success.

Returns mainly the following values in case of errors.

| Value | Description |
|---|---|
| SCE_LOCATION_ERROR_UNINITIALIZED | liblocation has not been initialized |

### Description

This function terminates liblocation. Execute this function once before unloading liblocation.

# sceLocationOpen

Start library operation

### Definition

```
#include <liblocation.h>
SceInt32 sceLocationOpen (
        SceLocationHandle *handle,
        SceUInt32 lmethod,
        SceUInt32 hmethod
)
```

### Arguments

**Out**

*handle* Pointer to variable for storing handle

**In**

*lmethod* Location calculation method

*hmethod* Direction calculation method

### Return Values

Returns SCE_LOCATION_SUCCESS for success.

Returns mainly the following values in case of errors.

| Value | Description |
|---|---|
| SCE_LOCATION_ERROR_INVALID_ADDRESS | Invalid handle storing address |
| SCE_LOCATION_ERROR_NO_MEMORY | Failed to allocate memory |
| SCE_LOCATION_ERROR_UNINITIALIZED | liblocation has not been initialized |
| SCE_LOCATION_ERROR_TOO_MANY_HANDLES | Could not be used because in use by many programs |
| SCE_LOCATION_ERROR_INVALID_LOCATION_METHOD | Specified location calculation method cannot be executed |
| SCE_LOCATION_ERROR_INVALID_HEADING_METHOD | Specified direction calculation method cannot be executed |
| SCE_LOCATION_ERROR_DISABLE_APPLICATION | Application cannot perform location calculation |
| SCE_LOCATION_ERROR_DISABLE_DEBUG_API | Debug APIs cannot be used |

### Description

This function acquires handles prior to liblocation use. If multiple programs simultaneously use the library, handles are used to distinguish users. Following handle obtainment, call functions using the obtained handles, and use sceLocationClose() to close the library.

Also specify the location and direction calculation methods in the arguments. The system has various calculation methods, and the method to be used for a calculation can be specified from among them.

Specify the logical OR of SceLocationLocationMethod and SceLocationLocationTiming in *lmethod*. Specify the logical OR of SceLocationHeading(Method/Timing/Resolution/Filter/NorthDirection/Stability) in *hmethod*.

liblocation allows the opening of 8 handles in the entire system. If 8 handles are already open due to other applications and other locations in the local application (including suspended applications) , `SCE_LOCATION_ERROR_TOO_MANY_HANDLES` will return and the use of liblocation will not be possible.

The dialog for obtaining permission to use location information may be displayed while this function is being executed. In such cases, execution will be blocked until the dialog is closed.

# sceLocationClose

Close library operation

### Definition

```
#include <liblocation.h>
SceInt32 sceLocationClose (
        SceLocationHandle handle
)
```

### Arguments

**In**
*handle* Handle

### Return Values

Returns SCE_LOCATION_SUCCESS for success.
Returns mainly the following values in case of errors.

| Value | Description |
|---|---|
| SCE_LOCATION_ERROR_INVALID_HANDLE | Invalid handle |

### Description

This function notifies the system that the library will no longer be used and disables the handles.

# sceLocationReopen

Reopen library operation

### Definition

```
#include <liblocation.h>
SceInt32 sceLocationReopen (
        SceLocationHandle handle,
        SceUInt32 lmethod,
        SceUInt32 hmethod
)
```

### Arguments

**In**

| | |
|---|---|
| *handle* | Handle |
| *lmethod* | Location calculation method |
| *hmethod* | Direction calculation method |

### Return Values

Returns SCE_LOCATION_SUCCESS for success.

Returns mainly the following values in case of errors. Even in the case of an error, execute sceLocationClose() when exiting.

| Value | Description |
|---|---|
| SCE_LOCATION_ERROR_INVALID_HANDLE | Invalid handle |
| SCE_LOCATION_ERROR_INVALID_LOCATION_METHOD | Specified location calculation method cannot be executed |
| SCE_LOCATION_ERROR_INVALID_HEADING_METHOD | Specified direction calculation method cannot be executed |
| SCE_LOCATION_ERROR_DISABLE_APPLICATION | Application cannot perform location calculation |
| SCE_LOCATION_ERROR_DISABLE_DEBUG_API | Debug APIs cannot be used |

### Description

This function reopens the specified handles in order to change the location or direction calculation method. Refer to sceLocationOpen() for an explanation on *lmethod* and *hmethod*.

If an error returns when this function is called, terminate liblocation by calling sceLocationClose().

The dialog for obtaining permission to use location information may be displayed while this function is being executed. In such cases, execution will be blocked until the dialog is closed.

# sceLocationGetMethod

Get location calculation method

### Definition

```
#include <liblocation.h>
SceInt32 sceLocationGetMethod (
        SceLocationHandle handle,
        SceUInt32 *lmethod,
        SceUInt32 *hmethod
)
```

### Arguments

**In**

*handle* Handle

**Out**

*lmethod* Pointer to variable for storing location calculation method. Specify NULL if not needed

*hmethod* Pointer to variable for storing direction calculation method. Specify NULL if not needed

### Return Values

Returns SCE_LOCATION_SUCCESS for success.

Returns mainly the following values in case of errors.

| Value | Description |
|---|---|
| SCE_LOCATION_ERROR_INVALID_HANDLE | Invalid handle |
| SCE_LOCATION_ERROR_DISABLE_APPLICATION | Application cannot perform location calculation |

### Description

Set NULL for the argument of calculation methods that need not be obtained.

This function gets the location and direction calculation methods that are currently being used. For the location calculation method, if multiple programs are using this library, the information that can be obtained with this function may differ from that specified with sceLocationOpen() or sceLocationReopen().

This function cannot obtain location information usage permission information.

# sceLocationGetLocation

Get location information

### Definition

```
#include <liblocation.h>
SceInt32 sceLocationGetLocation (
        SceLocationHandle handle,
        SceLocationLocationInfo *linfo
)
```

### Arguments

**In**

*handle* Handle

**Out**

*linfo* Pointer to variable for storing location information

### Return Values

Returns SCE_LOCATION_SUCCESS for success.

Returns mainly the following values in case of errors.

| Value | Description |
|---|---|
| SCE_LOCATION_ERROR_INVALID_HANDLE | Invalid handle |
| SCE_LOCATION_ERROR_INVALID_ADDRESS | Invalid data storing address |
| SCE_LOCATION_INFO_UNDETERMINED_LOCATION | Could not get information for location calculation |
| SCE_LOCATION_INFO_GET_LOCATION_CANCELED | sceLocationGetLocation() was canceled |
| SCE_LOCATION_INFO_DENIED_BY_USER | Location information usage is not allowed by the user |
| SCE_LOCATION_INFO_DISABLE_DEVICE | Location calculating device is off and location cannot be calculated |
| SCE_LOCATION_ERROR_DISABLE_APPLICATION | Application cannot perform location calculation |
| SCE_LOCATION_ERROR_MULTIPLE_GET_LOCATION | sceLocationGetLocation() or sceLocationGetLocationWithTimeout() is already being executed |

### Description

This function gets location information of the local device. This is a blocking-type function, and it may require ten or so seconds to return after it is called. To interrupt the processing, use the sceLocationCancelGetLocation() function from a different thread.

Since this function is designed to use a fixed timeout time, SCE_LOCATION_INFO_UNDETERMINED_LOCATION may often return as a return value in an environment where the GPS value is to be obtained. In this case, consider performing a retry or using the sceLocationGetLocationWithTimeout() function.

If SCE_LOCATION_MODE_CONFIRM_AUTO is specified in sceLocationInit()'s *mode*, the dialog for obtaining permission to use location information may be displayed before returning from this function.

# sceLocationGetLocationWithTimeout

Get location information (with timeout specified)

## Definition

```
#include <liblocation.h>
SceInt32 sceLocationGetLocationWithTimeout (
        SceLocationHandle handle,
        SceUInt32 timeout,
        SceLocationLocationInfo *linfo
)
```

## Arguments

**In**

*handle* Handle

*timeout* Timeout (seconds). If 0 is specified, this function continues to run until the value is obtained or it is cancelled

**Out**

*linfo* Pointer to variable for storing location information

## Return Values

Returns SCE_LOCATION_SUCCESS for success.

Returns mainly the following values in case of errors.

| Value | Description |
|---|---|
| SCE_LOCATION_ERROR_INVALID_HANDLE | Invalid handle |
| SCE_LOCATION_ERROR_INVALID_ADDRESS | Invalid data storing address |
| SCE_LOCATION_INFO_UNDETERMINED_LOCATION | Could not get information for location calculation |
| SCE_LOCATION_INFO_GET_LOCATION_CANCELED | sceLocationGetLocation() was canceled |
| SCE_LOCATION_INFO_DENIED_BY_USER | Location information usage is not allowed by the user |
| SCE_LOCATION_INFO_DISABLE_DEVICE | Location calculating device is off and location cannot be calculated |
| SCE_LOCATION_ERROR_DISABLE_APPLICATION | Application cannot perform location calculation |
| SCE_LOCATION_ERROR_MULTIPLE_GET_LOCATION | sceLocationGetLocation() or sceLocationGetLocationWithTimeout() is already being executed |

## Description

This function gets location information of the local device. In the case where it takes long time to get location information, this function can be used to adjust the wait time by setting timeout. Note that several minutes may be required to get the location information from GPS depending on the situation, so there is a possibility that the GPS value cannot be obtained if the wait time is not adjusted by, for example, setting longer timeout or accepting cancellation from the user.

This is a blocking-type function. To interrupt processing, use the sceLocationCancelGetLocation() function from a different thread.

If `SCE_LOCATION_MODE_CONFIRM_AUTO` is specified in `sceLocationInit()`'s *`mode`*, the dialog for obtaining permission to use location information may be displayed before returning from this function.

# sceLocationCancelGetLocation

Cancel location information obtainment operation

### Definition

```
#include <liblocation.h>
SceInt32 sceLocationCancelGetLocation (
        SceLocationHandle handle
)
```

### Arguments

**In**
*handle* Handle

### Return Values

Returns SCE_LOCATION_SUCCESS for success.

Returns mainly the following values in case of errors.

| Value | Description |
|---|---|
| SCE_LOCATION_ERROR_INVALID_HANDLE | Invalid handle |
| SCE_LOCATION_ERROR_DISABLE_APPLICATION | Application cannot perform location calculation |

### Description

This function cancels the obtainment of location information of the local device started with sceLocationGetLocation(). Location information obtainment may take several seconds, and this function can be used when a quick response, for example, for exiting the application, is required.

# sceLocationGetLocationWifiHistory

Get location information (with stored Wi-Fi access points obtainment)

### Definition

```
#include <liblocation.h>
SceInt32 sceLocationGetLocationWifiHistory (
        SceLocationHandle handle,
        SceUInt32 timeout,
        SceLocationLocationInfo *linfo
)
```

### Arguments

**In**

*handle* Handle

*timeout* Timeout (seconds). If 0 is specified, this function continues to run until the value is obtained or it is cancelled

**Out**

*linfo* Pointer to variable for storing location information

### Return Values

Returns SCE_LOCATION_SUCCESS for success.

Returns mainly the following values in case of errors.

| Value | Description |
|---|---|
| SCE_LOCATION_ERROR_INVALID_HANDLE | Invalid handle |
| SCE_LOCATION_ERROR_INVALID_ADDRESS | Invalid data storing address |
| SCE_LOCATION_INFO_UNDETERMINED_LOCATION | Information to calculate location cannot be obtained |
| SCE_LOCATION_INFO_NOT_FOUND_AP | Access points could not be found. Wi-Fi access point information will not be stored |
| SCE_LOCATION_INFO_GET_LOCATION_CANCELED | sceLocationGetLocationWifiHistory() was canceled |
| SCE_LOCATION_INFO_DENIED_BY_USER | Location information usage is not allowed by the user |
| SCE_LOCATION_INFO_DISABLE_DEVICE | Location calculating device is off and location cannot be calculated |
| SCE_LOCATION_ERROR_DISABLE_APPLICATION | Application cannot perform location calculation |
| SCE_LOCATION_ERROR_MULTIPLE_GET_LOCATION | sceLocationGetLocation(), sceLocationGetLocationWithTimeout() or sceLocationGetLocationWifiHistory() is already being executed |

## Description

This function gets location information of the local device.

When `SCE_LOCATION_INFO_UNDETERMINED_LOCATION` returns, only information regarding the surrounding Wi-Fi access points are stored within liblocation. Once network connection becomes possible, the location information corresponding to the Wi-Fi access point information held within liblocation can be obtained using `sceLocationQueryLocationWifiHistory()`. Although the number of data that can be stored depends on the usage condition of other applications, the maximum is several hundred points.

When the current location can be calculated or when surrounding Wi-Fi access points cannot be detected, data will not be stored. Other features of this function are the same as `sceLocationGetLocationWithTimeout()`.

This is a blocking-type function. To interrupt processing, use the `sceLocationCancelGetLocation()` function from a different thread.

If `SCE_LOCATION_MODE_CONFIRM_AUTO` is specified in `sceLocationInit()`'s *`mode`*, the dialog for obtaining permission to use location information may be displayed before returning from this function.

## Notes

This function can only be used when `SCE_LOCATION_MODE_V3` or later is specified to *`mode`* of `sceLocationInit()`.

# sceLocationQueryLocationWifiHistory

Send query for location information from the stored Wi-Fi access points

## Definition

```
#include <liblocation.h>
SceInt32 sceLocationQueryLocationWifiHistory (
        SceLocationHandle handle,
        SceRtcTick timestamp,
        SceUInt32 timeout,
        SceLocationLocationInfo *linfo
)
```

## Arguments

**In**

| | |
|---|---|
| *handle* | Handle |
| *timestamp* | *timestamp* obtained with sceLocationGetLocationWifiHistory() |
| *timeout* | Timeout (seconds). If 0 is specified, this function continues to run until the value is obtained or it is cancelled |

**Out**

| | |
|---|---|
| *linfo* | Pointer to variable for storing location information |

## Return Values

Returns SCE_LOCATION_SUCCESS for success.

Returns mainly the following values in case of errors.

| Value | Description |
|---|---|
| SCE_LOCATION_ERROR_INVALID_HANDLE | Invalid handle |
| SCE_LOCATION_ERROR_INVALID_ADDRESS | Invalid data storing address |
| SCE_LOCATION_INFO_INVALID_TIMESTAMP | Specified *timestamp* is not registered in the stored Wi-Fi access points |
| SCE_LOCATION_INFO_NETWORK_NOT_ACCESSED | Calculation will not be carried out because there is no network connection |
| SCE_LOCATION_INFO_GET_LOCATION_CANCELED | sceLocationQueryLocationWifiHistory() was canceled |
| SCE_LOCATION_INFO_AP_NOT_IN_LOCATION_DB | Specified access point is not stored in location database |
| SCE_LOCATION_INFO_TIMEOUT | Calculation incomplete due to timeout |
| SCE_LOCATION_INFO_DENIED_BY_USER | Location information usage is not allowed by the user |
| SCE_LOCATION_ERROR_DISABLE_APPLICATION | Application cannot perform location calculation |
| SCE_LOCATION_ERROR_MULTIPLE_GET_LOCATION | sceLocationGetLocation(), sceLocationGetLocationWithTimeout() or sceLocationGetLocationWifiHistory() is already being executed |

### Description

This function accesses the stored Wi-Fi access points obtained with `sceLocationGetLocationWifiHistory()` to obtain location calculation information for an earlier location.

*`timestamp`* obtained with `sceLocationGetLocationWifiHistory()` is used as the query key. This *`timestamp`* is only valid for the application that executed `sceLocationGetLocationWifiHistory()`.

There is no guarantee that data stored with `sceLocationGetLocationWifiHistory()` will be kept, as the function (or another application) may add new Wi-Fi access points or the data may be implicitly deleted after a certain time duration.

If `SCE_LOCATION_MODE_CONFIRM_AUTO` is specified in `sceLocationInit()`'s *`mode`*, the dialog for obtaining permission to use location information may be displayed before returning from this function.

### Notes

This function can only be used when `SCE_LOCATION_MODE_V3` or later is specified to *`mode`* of `sceLocationInit()`.

# sceLocationCancelQueryLocationWifiHistory

Cancel query for location information from the stored Wi-Fi access points

## Definition

```
#include <liblocation.h>
SceInt32 sceLocationCancelQueryLocationWifiHistory (
        SceLocationHandle handle
)
```

## Arguments

**In**

*handle* Handle

## Return Values

Returns SCE_LOCATION_SUCCESS for success.

Returns mainly the following values in case of errors.

| Value | Description |
|---|---|
| SCE_LOCATION_ERROR_INVALID_HANDLE | Invalid handle |
| SCE_LOCATION_ERROR_DISABLE_APPLICATION | Application cannot perform location calculation |

## Description

This function cancels the operation started by sceLocationQueryLocationWifiHistory() to obtain location information of the local device. Location information obtainment may take several seconds; this function can be used when an immediate response is required, for example, upon application termination.

## Notes

This function can only be used when SCE_LOCATION_MODE_V3 or later is specified to *mode* of sceLocationInit().

# sceLocationStartLocationCallback

Start continual obtainment of location information

### Definition

```
#include <liblocation.h>
SceInt32 sceLocationStartLocationCallback (
        SceLocationHandle handle,
        SceUInt32 distance,
        SceLocationLocationInfoCallback callback,
        void *userdata
)
```

### Arguments

**In**

| | |
|---|---|
| *handle* | Handle |
| *distance* | Notify callback only when the distance from the location notified by the previous callback equals or exceeds this distance (meters). All calculation results will be notified if 0 is specified |
| *callback* | Pointer to location information callback function |
| *userdata* | Pointer to userdata to be notified to callback. Specify NULL if not needed |

### Return Values

Returns SCE_LOCATION_SUCCESS for success.

Returns mainly the following values in case of errors.

| Value | Description |
|---|---|
| SCE_LOCATION_ERROR_INVALID_HANDLE | Invalid handle |
| SCE_LOCATION_ERROR_INVALID_ADDRESS | Invalid callback address |
| SCE_LOCATION_ERROR_NO_MEMORY | Failed to allocate memory |
| SCE_LOCATION_ERROR_MULTIPLE_CALLBACK | Callback already started |
| SCE_LOCATION_ERROR_DISABLE_APPLICATION | Application cannot perform location calculation |

### Description

This function starts continual obtainment of location information of the local device.

If *distance* is set to a value other than 0, callback returns if the difference with the previous value is equal to or greater than *distance* (meters), even if it is shorter than the timing (SceLocationLocationTiming) specified with *lmethod* of sceLocationOpen().

# sceLocationStopLocationCallback

Stop continual obtainment of location information

### Definition

```
#include <liblocation.h>
SceInt32 sceLocationStopLocationCallback (
        SceLocationHandle handle
)
```

### Arguments

**In**
*handle* Handle

### Return Values

Returns SCE_LOCATION_SUCCESS for success.

Returns mainly the following values in case of errors.

| Value | Description |
|---|---|
| SCE_LOCATION_ERROR_INVALID_HANDLE | Invalid handle |
| SCE_LOCATION_ERROR_NOT_RUNNING_CALLBACK | Callback has not started |
| SCE_LOCATION_ERROR_DISABLE_APPLICATION | Application cannot perform location calculation |

### Description

This function stops continual obtainment of location information of the local device. Once this function normally ends, callback will not be called for the handle.

To execute sceLocationClose() or sceLocationReopen() while callback execution is in progress, be sure to execute this function beforehand in order to stop the callback.

# sceLocationGetHeading

Get direction information

### Definition

```
#include <liblocation.h>
SceInt32 sceLocationGetHeading (
        SceLocationHandle handle,
        SceLocationHeadingInfo *hinfo
)
```

### Arguments

**In**

*handle* Handle

**Out**

*hinfo* Pointer to variable for storing direction information

### Return Values

Returns SCE_LOCATION_SUCCESS for success.

Returns mainly the following values in case of errors.

| Value | Description |
|---|---|
| SCE_LOCATION_ERROR_INVALID_HANDLE | Invalid handle |
| SCE_LOCATION_ERROR_INVALID_ADDRESS | Invalid data storing address |
| SCE_LOCATION_INFO_UNDETERMINED_HEADING | Could not get direction information |

### Description

This function gets direction information of the local device. Since the magnetometer sensor device is initialized immediately after the execution of sceLocationOpen(), SCE_LOCATION_INFO_UNDETERMINED_HEADING returns as a return value, and direction information may not be obtained in some cases. Also, the magnetometer sensor device is unstable for a while even after a value is successfully obtained. Therefore, it is recommended to execute this function at least 500 milliseconds to 1 second after the execution of sceLocationOpen().

# sceLocationStartHeadingCallback

Start continual obtainment of direction information

### Definition

```
#include <liblocation.h>
SceInt32 sceLocationStartHeadingCallback (
        SceLocationHandle handle,
        SceFloat32 difference,
        SceLocationHeadingInfoCallback callback,
        void *userdata
)
```

### Arguments

**In**

| | |
|---|---|
| *handle* | Handle |
| *difference* | Notify callback only when the distance from the direction notified during the previous callback is larger than this angle. Specify a value in the range of 0 to 180.0. All calculation results will be notified if 0 is specified |
| *callback* | Pointer to location information callback function |
| *userdata* | Pointer to userdata to be notified to callback. Specify NULL if not needed |

### Return Values

Returns SCE_LOCATION_SUCCESS for success.

Returns mainly the following values in case of errors.

| Value | Description |
|---|---|
| SCE_LOCATION_ERROR_INVALID_HANDLE | Invalid handle |
| SCE_LOCATION_ERROR_INVALID_ADDRESS | Invalid callback address |
| SCE_LOCATION_ERROR_NO_MEMORY | Failed to allocate memory |
| SCE_LOCATION_ERROR_MULTIPLE_CALLBACK | Callback already started |
| SCE_LOCATION_ERROR_INVALID_VALUE | An invalid value was entered |

### Description

This function starts continual obtainment of direction information of the local device.

If *difference* is set to a value other than 0, callback returns if the difference with the previous value is equal to or greater than *difference*, even if it is shorter than the timing (SceLocationHeadingTiming) specified with *hmethod* of sceLocationOpen().

Since the magnetometer sensor device is initialized immediately after the execution of sceLocationOpen(), SCE_LOCATION_INFO_UNDETERMINED_HEADING returns to *result* of the callback function, and a value may not be obtained in some cases. Also, the magnetometer sensor device is unstable for a while even after a value is successfully obtained. Therefore, it is recommended to execute this function at least 500 milliseconds to 1 second after the execution of sceLocationOpen().

# sceLocationStopHeadingCallback

Stop continual obtainment of direction information

## Definition

```
#include <liblocation.h>
SceInt32 sceLocationStopHeadingCallback (
        SceLocationHandle handle
)
```

## Arguments

**In**

*handle* Handle

## Return Values

Returns SCE_LOCATION_SUCCESS for success.

Returns mainly the following values in case of errors.

| Value | Description |
|---|---|
| SCE_LOCATION_ERROR_INVALID_HANDLE | Invalid handle |
| SCE_LOCATION_ERROR_NOT_RUNNING_CALLBACK | Callback has not started |

## Description

This function stops continual obtainment of direction information of the local device. Once this function normally ends, callback will not be called for the handle.

To execute sceLocationClose() or sceLocationReopen() while callback execution is in progress, be sure to execute this function beforehand in order to stop the callback.

# sceLocationConfirm

Display dialog for obtaining permission to use location information

### Definition

```
#include <liblocation.h>
SceInt32 sceLocationConfirm (
        SceLocationHandle handle
)
```

### Arguments

**In**

*handle* Handle

### Return Values

Returns SCE_LOCATION_SUCCESS for success.

Returns mainly the following values in case of errors.

| Value | Description |
|---|---|
| SCE_LOCATION_ERROR_INVALID_HANDLE | Invalid handle |
| SCE_LOCATION_ERROR_DISABLE_APPLICATION | Application cannot perform location calculation |

### Description

This function displays a dialog box for requesting location information usage permission, prompting the user to allow location information usage, and reflects the result to liblocation, the system settings, etc.

To get location information for the first time after the application has been installed, or if location data or application-specific service has not been set in the system settings, execute a location information-related function such as sceLocationGetLocation(), SceLocationLocationInfoCallback(), or sceLocationGetMethod(), and if an error such as SCE_LOCATION_INFO_DENIED_BY_USER is returned, execute the sceLocationConfirm() function to obtain the location information usage permission. However, usage permission through parental control cannot be changed. Furthermore, if this function is executed when all usage permissions have been obtained, SCE_LOCATION_SUCCESS will return.

# sceLocationConfirmGetStatus

Get status of dialog for obtaining permission to use location information

## Definition

```
#include <liblocation.h>
SceInt32 sceLocationConfirmGetStatus (
        SceLocationHandle handle,
        SceLocationDialogStatus *status
)
```

## Arguments

**In**

*handle* Handle

**Out**

*status* Current dialog display status

## Return Values

Returns SCE_LOCATION_SUCCESS for success.

Returns mainly the following values in case of errors.

| Value | Description |
|---|---|
| SCE_LOCATION_ERROR_INVALID_HANDLE | Invalid handle |
| SCE_LOCATION_ERROR_DISABLE_APPLICATION | Application cannot perform location calculation |

## Description

This function gets the execution status of the dialog currently displayed with sceLocationConfirm().

# sceLocationConfirmGetResult

Get result of dialog for obtaining permission to use location information

### Definition

```
#include <liblocation.h>
SceInt32 sceLocationConfirmGetResult (
        SceLocationHandle handle,
        SceLocationDialogResult *result
)
```

### Arguments

**In**

*handle* Handle

**Out**

*result* Result obtained through dialog

### Return Values

Returns SCE_LOCATION_SUCCESS for success.

Returns mainly the following values in case of errors.

| Value | Description |
|---|---|
| SCE_LOCATION_ERROR_INVALID_HANDLE | Invalid handle |
| SCE_LOCATION_ERROR_DIALOG_RESULT_NONE | Could not get dialog result |
| SCE_LOCATION_ERROR_DISABLE_APPLICATION | Application cannot perform location calculation |

### Description

This function gets the result of the dialog displayed with sceLocationConfirm(). When the result of status obtainment with sceLocationConfirmGetStatus() is SCE_LOCATION_DIALOG_STATUS_IDLE or SCE_LOCATION_DIALOG_STATUS_RUNNING, result obtainment is not possible and SCE_LOCATION_ERROR_DIALOG_RESULT_NONE is output as the return value.

# sceLocationConfirmAbort

Abort dialog for obtaining permission to use location information

### Definition

```
#include <liblocation.h>
SceInt32 sceLocationConfirmAbort (
        SceLocationHandle handle
)
```

### Arguments

**In**
*handle* Handle

### Return Values

Returns SCE_LOCATION_SUCCESS for success.
Returns mainly the following values in case of errors.

| Value | Description |
|---|---|
| SCE_LOCATION_ERROR_INVALID_HANDLE | Invalid handle |
| SCE_LOCATION_ERROR_DISABLE_APPLICATION | Application cannot perform location calculation |

### Description

This function is used to abort and delete the dialog displayed with sceLocationConfirm(). If sceLocationConfirmGetStatus() is executed after executing this function, SCE_LOCATION_DIALOG_STATUS_IDLE is output.

# sceLocationGetPermission

Get location information usage permission/refusal information

## Definition

```
#include <liblocation.h>
SceInt32 sceLocationGetPermission (
        SceLocationHandle handle,
        SceLocationPermissionInfo *info
)
```

## Arguments

**In**

*handle* Handle

**Out**

*info* Detailed location information usage permission information. Specify NULL if this is not necessary

## Return Values

Returns SCE_LOCATION_SUCCESS when usage of location information is allowed.

Returns mainly the following values in case of errors or when usage of location information is denied.

| Value | Description |
|---|---|
| SCE_LOCATION_ERROR_INVALID_HANDLE | Invalid handle |
| SCE_LOCATION_ERROR_DISABLE_APPLICATION | Application cannot perform location calculation |
| SCE_LOCATION_INFO_DENIED_BY_USER | Usage of location information denied by the user. |

## Description

This function gets the current location information usage permission status, which is a result of various causes. If you wish to know whether the usage of information is allowed before using location information, execute this function and check whether the return value is SCE_LOCATION_INFO_DENIED_BY_USER. Refer to *info* if you wish to obtain detailed permission information.

# sceLocationDenyApplication

Deny location information usage permission of this application

## Definition

```
#include <liblocation.h>
SceInt32 sceLocationDenyApplication (
        SceLocationHandle handle
)
```

## Arguments

**In**
*handle* Handle

## Return Values

Returns SCE_LOCATION_SUCCESS for success.

Returns mainly the following values in case of errors.

| Value | Description |
|---|---|
| SCE_LOCATION_ERROR_INVALID_HANDLE | Invalid handle |
| SCE_LOCATION_ERROR_DISABLE_APPLICATION | Application cannot perform location calculation |

## Description

This function denies the usage permission of location information for location data items in system settings by the application. The usage permission will be denied without displaying a dialog box to notify this setting.

# sceLocationSetThreadParameter

Change thread settings of liblocation

### Definition

```
#include <liblocation.h>
SceInt32 sceLocationSetThreadParameter (
        SceInt32 intPriority,
        SceInt32 cpuAffinityMask
)
```

### Arguments

**In**

| | |
|---|---|
| *intPriority* | Thread priority |
| *cpuAffinityMask* | CPU affinity mask |

### Return Values

Returns `SCE_LOCATION_SUCCESS` for success.

Returns mainly the following values in case of errors.

| Value | Description |
|---|---|
| `SCE_LOCATION_ERROR_INITIALIZED` | liblocation has been initialized |

### Description

This function sets the thread priority and affinity mask used in liblocation. Execute this function before executing `sceLocationInit()`.

# sceLocationSetGpsEmulationFile

Specify GPS emulation file

### Definition

```
#include <liblocation.h>
SceInt32 sceLocationSetGpsEmulationFile (
        unsigned char *filename
)
```

### Arguments

**In**

*filename* Emulation file name

### Return Values

Returns SCE_LOCATION_SUCCESS for success.

Returns mainly the following values in case of errors.

| Value | Description |
|---|---|
| SCE_LOCATION_ERROR_INVALID_ADDRESS | Invalid data storing address |
| SCE_LOCATION_ERROR_DISABLE_DEBUG_API | Debug APIs cannot be used |
| SCE_LOCATION_DBGERROR_EMULATION_FILE_ERROR | Specified file for GPS emulation not found |

### Description

liblocation performs GPS emulation during application development and can be used for development and debugging of devices not equipped with GPS or in locations where radio waves do not penetrate. This function specifies the file that is used for GPS emulation as a filename with the absolute path from the root directory of the memory card. However, owing to file system limitations, only locations under ux0:/data/gpsdata/ can be used. If GPS emulation is performed without executing this API, ux0:/data/gpsdata/gspdata.nma will be selected as default. The specified file will become valid when sceLocationOpen() is performed, and continue until sceLocationOpen() is executed specifying a new filename. Also, this setting will be valid for the entire system, and will also be reflected on other applications running simultaneously. For details, refer to the "GPS Emulation Function" chapter in "liblocation Overview".

An error will return at the time of application release.

# Constants

# SceLocationMode

liblocation mode

### Definition

| Value | (Number) | Description |
|---|---|---|
| SCE_LOCATION_MODE_V1 | 0x00000000 | Mode with compatibility to liblocation of versions SDK 1.690 and earlier. |
| SCE_LOCATION_MODE_V2 | 0x00000001 | SCE_LOCATION_INFO_DISABLE_DEVICE output mode. |
| SCE_LOCATION_MODE_V3 | 0x00000002 | Mode in which all features can be used. Use this mode as a general rule. |

### Description

In order to maintain liblocation compatibility and in order to lessen unnecessary resource consumption, set the optimized mode specified within liblocation. Specify this constant to *mode* of sceLocationInit().

SCE_LOCATION_MODE_V1 is a mode in which APIs and constants of SDK versions 1.690 and earlier can be used.

SCE_LOCATION_MODE_V2 is a mode with features of SCE_LOCATION_MODE_V1, as well as a feature to output the SCE_LOCATION_INFO_DISABLE_DEVICE error code.

As a general rule, specify SCE_LOCATION_MODE_V3 in which all features of liblocation can be used.

Programs built without the possibility of the SCE_LOCATION_INFO_DISABLE_DEVICE error code (added from SDK 1.800) returning should use SCE_LOCATION_MODE_V1.

To save memory consumption, specify SCE_LOCATION_MODE_V2 when functions for SCE_LOCATION_MODE_V3 are not used.

When specifying SCE_LOCATION_MODE_V3, specify SCE_SYSMODULE_LOCATION_EXTENSION in sceSysmoduleLoadModule(), and upon building the program, link libSceLocation_stub.a and libSceLocationExtension_stub.a.

# SceLocationModeConfirm

Mode for obtaining permission to use location information

## Definition

| Value | (Number) | Description |
|---|---|---|
| SCE_LOCATION_MODE_CONFIRM_MANUAL | 0x00000000 | Obtains permission to use location information by using checking APIs |
| SCE_LOCATION_MODE_CONFIRM_AUTO | 0x00000100 | Obtains permission to use location information automatically |

## Description

SCE_LOCATION_MODE_CONFIRM_AUTO is a mode whereby permission to use location information is obtained automatically. By using this mode, the system will display the dialog for obtaining permission to use location information at the necessary times, and will ask the end user for permission to use location information. Basically, the application does not require implementation for obtaining permission.

SCE_LOCATION_MODE_CONFIRM_MANUAL is a mode using APIs for obtaining permission to use location information, such as sceLocationConfirm(). Use this mode if you wish to control the timing in which the dialog for obtaining permission to use location information is displayed. Also, if you specify this mode, the dialog for obtaining permission to use location information may be displayed while sceLocationOpen() is being executed, in which case sceLocationOpen() will be blocked. The dialog will not be displayed while sceLocationOpen() is being executed if the application is built on a version preceding SDK 2.100.

# SceLocationLocationMethod

Location calculating device

### Definition

| Value | (Number) | Description |
|---|---|---|
| SCE_LOCATION_LMETHOD_NONE | 0x00000000 | Do not perform location calculation |
| SCE_LOCATION_LMETHOD_AGPS_AND_3G_AND_WIFI | 0x00000001 | Perform location calculation by switching automatically among AGPS, Wi-Fi, and 3G |
| SCE_LOCATION_LMETHOD_3G_AND_WIFI | 0x00000002 | Perform location calculation by switching automatically between 3G and Wi-Fi |
| SCE_LOCATION_LMETHOD_WIFI | 0x00000003 | Use only Wi-Fi |
| SCE_LOCATION_LMETHOD_3G | 0x00000004 | Use only 3G |
| SCE_LOCATION_LMETHOD_GPS | 0x00000005 | Use only GPS |
| SCE_LOCATION_LMETHOD_GPS_EMULATION | 0x0000000F | Perform GPS emulation |
| SCE_LOCATION_LMETHOD_MASK | 0x0000000F | Mask for obtaining the location calculating device |

### Description

These constants represent the location calculating device.

If multiple programs are using this library, calculation might not be performed with the device that has been set. In order to verify the current usage status of devices, it is necessary to obtain the value with `sceLocationGetMethod()` and extract the value using `SCE_LOCATION_LMETHOD_MASK`.

Depending on the set, it will not be possible to use some devices. In this case, an error will not be returned, but the device will not function and a substitute device will perform the calculation.

While in `SCE_LOCATION_LMETHOD_GPS_EMULATION` mode, if any files specified with `sceLocationSetGpsEmulationFile()`, etc. are present on the memory card, etc. GPS emulation will be performed. An error will return if `SCE_LOCATION_LMETHOD_GPS_EMULATION` is set at the time of application release.

# SceLocationLocationTiming

Interval of location calculation

### Definition

| Value | (Number) | Description |
|---|---|---|
| SCE_LOCATION_LOCATION_TIMING_MINIMUM | 0x00000000 | Calculates with minimum intervals |
| SCE_LOCATION_LOCATION_TIMING_5SEC | 0x00000010 | Calculates with 5 second intervals |
| SCE_LOCATION_LOCATION_TIMING_10SEC | 0x00000020 | Calculates with 10 second intervals |
| SCE_LOCATION_LOCATION_TIMING_30SEC | 0x00000030 | Calculates with 30 second intervals |
| SCE_LOCATION_LOCATION_TIMING_60SEC | 0x00000040 | Calculates with 60 second intervals |
| SCE_LOCATION_LOCATION_TIMING_10MIN | 0x00000050 | Calculates with 10 minute intervals |
| SCE_LOCATION_LOCATION_TIMING_MASK | 0x00000070 | Mask for obtaining calculation interval |

### Description

These constants represent the minimum interval for internal calculations during location calculation. During location calculation, power consumption is inversely proportional to the length of internal calculation intervals. Therefore, it is advisable to set intervals in accordance with the usage status of each application.

If multiple programs are using this library, the timing of location calculation may differ from settings. In order to verify current calculation intervals, it is necessary to obtain the value with `sceLocationGetMethod()` and `SCE_LOCATION_LOCATION_TIMING_MASK`.

This value only regulates the timing of internal calculation, and output will not necessarily follow this timing. Therefore, this value cannot be used as a timer. Furthermore, this interval will not influence `sceLocationGetLocation()`.

# SceLocationHeadingMethod

Direction output reference

### Definition

| Value | (Number) | Description |
|---|---|---|
| SCE_LOCATION_HMETHOD_NONE | 0x00000000 | Do not perform direction calculation |
| SCE_LOCATION_HMETHOD_AUTO | 0x00000001 | Automatically determine hold orientation and outputs its value |
| SCE_LOCATION_HMETHOD_VERTICAL | 0x00000002 | Output value in vertical hold reference system |
| SCE_LOCATION_HMETHOD_HORIZONTAL | 0x00000003 | Output value in horizontal hold reference system |
| SCE_LOCATION_HMETHOD_CAMERA | 0x00000004 | Output value in camera axis reference system |
| SCE_LOCATION_HMETHOD_MASK | 0x00000007 | Mask for obtaining output reference system |

### Description

These constants represent the reference system of direction output. When direction is calculated internally, calculation error may become larger depending on the local device's position. It is therefore necessary to choose a suitable reference system for the position. SCE_LOCATION_HMETHOD_AUTO automatically determines vertical or horizontal hold through the system, and performs internal calculations in accordance with the position. This, however, will not result in selection of the camera axis reference system. Basically, choose SCE_LOCATION_HMETHOD_AUTO when not selecting the camera axis reference system.

# SceLocationHeadingTiming

Direction calculation interval

### Definition

| Value | (Number) | Description |
|---|---|---|
| SCE_LOCATION_HEADING_TIMING_MINIMUM | 0x00000000 | Calculates with minimum intervals |
| SCE_LOCATION_HEADING_TIMING_64MS | 0x00000010 | Calculates with intervals of approx. 64 ms |
| SCE_LOCATION_HEADING_TIMING_100MS | 0x00000020 | Calculates with intervals of approx. 100 ms |
| SCE_LOCATION_HEADING_TIMING_500MS | 0x00000030 | Calculates with intervals of approx. 500 ms |
| SCE_LOCATION_HEADING_TIMING_1SEC | 0x00000040 | Calculates with intervals of approx. 1 second |
| SCE_LOCATION_HEADING_TIMING_MASK | 0x00000070 | Mask for obtaining calculation interval |

### Description

These constants represent the minimum interval for internal calculations during direction calculation. This can be used, for example, when callbacks are too frequent with minimum intervals.

This value only regulates the timing of internal calculations, and output will not necessarily follow this timing. Therefore, this value cannot be used as a timer. Furthermore, this interval will not influence `sceLocationGetHeading()`.

# SceLocationHeadingResolution

Direction calculation boundary

### Definition

| Value | (Number) | Description |
|---|---|---|
| SCE_LOCATION_HEADING_RESOLUTION_MINIMUM | 0x00000000 | Outputs in minimum units |
| SCE_LOCATION_HEADING_RESOLUTION_DEG1 | 0x00000100 | Outputs in 1 degree units |
| SCE_LOCATION_HEADING_RESOLUTION_DEG5 | 0x00000200 | Outputs in 5 degree units |
| SCE_LOCATION_HEADING_RESOLUTION_DIV32 | 0x00000300 | Outputs in 32 divisions of 360 degrees |
| SCE_LOCATION_HEADING_RESOLUTION_DIV16 | 0x00000400 | Outputs in 16 divisions of 360 degrees |
| SCE_LOCATION_HEADING_RESOLUTION_DIV8 | 0x00000500 | Outputs in 8 divisions of 360 degrees |
| SCE_LOCATION_HEADING_RESOLUTION_DIV4 | 0x00000600 | Outputs in 4 divisions of 360 degrees |
| SCE_LOCATION_HEADING_RESOLUTION_MASK | 0x00000700 | Mask for obtaining calculation boundary |

### Description

During direction calculation, outputs are made by rounding the value to the unit specified by these constants. Other than the minimum unit, if SCE_LOCATION_HEADING_FILTER_HYSTERESIS_ENABLE is specified, output will be performed with hysteresis.

# SceLocationHeadingFilter

Direction filter

### Definition

| Value | (Number) | Description |
|---|---|---|
| SCE_LOCATION_HEADING_FILTER_AVERAGE_DISABLE | 0x00000000 | Averaging filter disabled |
| SCE_LOCATION_HEADING_FILTER_AVERAGE_ENABLE | 0x00001000 | Averaging filter enabled |
| SCE_LOCATION_HEADING_FILTER_AVERAGE_MASK | 0x00001000 | Mask for obtaining averaging filter status |

| Value | (Number) | Description |
|---|---|---|
| SCE_LOCATION_HEADING_FILTER_HYSTERESIS_DISABLE | 0x00000000 | Hysteresis filter disabled |
| SCE_LOCATION_HEADING_FILTER_HYSTERESIS_ENABLE | 0x00002000 | Hysteresis filter enabled |
| SCE_LOCATION_HEADING_FILTER_HYSTERESIS_MASK | 0x00002000 | Mask for obtaining hysteresis filter status |

### Description

When outputting direction, these direction filters are used to adjust the output value. It is also possible to use filter algorithms simultaneously.

# SceLocationHeadingNorthDirection

Set north direction

### Definition

| Value | (Number) | Description |
|---|---|---|
| SCE_LOCATION_HEADING_NORTH_TRUE | 0x00000000 | Indicates true north |
| SCE_LOCATION_HEADING_NORTH_MAGNETIC | 0x80000000 | Indicates magnetic north |
| SCE_LOCATION_HEADING_NORTH_MASK | 0x80000000 | Mask for obtaining north direction settings |

### Description

These constants set the reference system for the north direction. In the case of SCE_LOCATION_HEADING_NORTH_TRUE, declination will be inferred and adjusted based on calculated location information.

# SceLocationHeadingStability

Stability of direction information

### Definition

| Value | (Number) | Description |
|---|---|---|
| SCE_LOCATION_HEADING_STABILITY_UNSTABLE | 0x00000000 | Status of magnetometer sensor being unstable |
| SCE_LOCATION_HEADING_STABILITY_STABLE | 0x00000001 | Status of magnetometer sensor being stable (not output in this version) |
| SCE_LOCATION_HEADING_STABILITY_VERYSTABLE | 0x00000002 | Status of magnetometer sensor being very stable |

### Description

This indicates the status of the magnetometer sensor.

SCE_LOCATION_HEADING_STABILITY_UNSTABLE is a status in which the stability of the magnetometer sensor is poor and calibration assistance is required.
SCE_LOCATION_HEADING_STABILITY_STABLE and SCE_LOCATION_HEADING_STABILITY_VERYSTABLE are statuses in which calibration assistance is not required. (In this version, SCE_LOCATION_HEADING_STABILITY_STABLE is not output.)

# SceLocationDialogStatus

Display status of dialog for obtaining permission to use location information

### Definition

| Value | (Number) | Description |
|---|---|---|
| SCE_LOCATION_DIALOG_STATUS_IDLE | 0 | Dialog initial idle status |
| SCE_LOCATION_DIALOG_STATUS_RUNNING | 1 | Dialog running |
| SCE_LOCATION_DIALOG_STATUS_FINISHED | 2 | Dialog operation finished |

### Description

This constant represents the display status of the dialog for obtaining permission to use location information obtained with `sceLocationConfirmGetStatus()`. When the dialog status obtainment result is `SCE_LOCATION_DIALOG_STATUS_IDLE` or `SCE_LOCATION_DIALOG_STATUS_RUNNING`, Obtaining result of the dialog for obtaining permission to use location information is not possible and `SCE_LOCATION_DIALOG_RESULT_NONE` is output.

# SceLocationDialogResult

Result of dialog for obtaining permission to use location information

## Definition

| Value | (Number) | Description |
|---|---|---|
| SCE_LOCATION_DIALOG_RESULT_NONE | 0 | Result is not stored |
| SCE_LOCATION_DIALOG_RESULT_DISABLE | 1 | Negative result is stored |
| SCE_LOCATION_DIALOG_STATUS_ENABLE | 2 | Positive result is stored |

## Description

This constant represents the result of the dialog for obtaining permission to use location information obtained with `sceLocationConfirmGetResult()`.

# SceLocationPermissionApplicationStatus

liblocation usage permission status for individual application

## Definition

| Value | (Number) | Description |
|---|---|---|
| SCE_LOCATION_PERMISSION_APPLICATION_NONE | 0 | liblocation not used |
| SCE_LOCATION_PERMISSION_APPLICATION_INIT | 1 | liblocation not accessed |
| SCE_LOCATION_PERMISSION_APPLICATION_DENY | 2 | liblocation access denied status |
| SCE_LOCATION_PERMISSION_APPLICATION_ALLOW | 3 | liblocation access allowed status |

## Description

This constant represents the usage permission status of each application, such as liblocation access allowed/denied status.

# SceLocationPermissionStatus

liblocation usage permission status

## Definition

| Value | (Number) | Description |
|---|---|---|
| SCE_LOCATION_PERMISSION_DENY | 0 | liblocation access denied status |
| SCE_LOCATION_PERMISSION_ALLOW | 1 | liblocation access allowed status |

## Description

This constant represents the usage permission status, such as liblocation access allowed/denied status.

# SceLocationPermissionUpdate

liblocation usage permission status update status

### Definition

| Value | (Number) | Description |
|---|---|---|
| SCE_LOCATION_PERMISSION_NOT_UPDATED | 0x00000000 | The contents of SceLocationPermissionInfo have not been changed |
| SCE_LOCATION_PERMISSION_UPDATED | 0x00000001 | The contents of SceLocationPermissionInfo have been changed |

### Description

These constants indicate whether the contents of SceLocationPermissionInfo have been changed after sceLocationConfirm() was executed.

# Other Constants

Specific data value

**Definition**

| Value | (Number) | Description |
|---|---|---|
| SCE_LOCATION_DATA_INVALID | -9999.0 | Value stored when value could not be obtained |
| SCE_LOCATION_HANDLE_INVALID | 0xFFFFFFFF | Invalid handle value |

# Error Codes

# Error Codes

List of error codes returned by liblocation

## Definition

### No error

The following error codes indicate function success or a special state. Debugging is not required.

| Value | (Number) | Description |
|---|---|---|
| SCE_LOCATION_SUCCESS | 0x00000000 | Successful |
| SCE_LOCATION_INFO_UNDETERMINED_LOCATION | 0x80101200 | Could not get information for location calculation |
| SCE_LOCATION_INFO_INSUFFICIENT_INFORMATION | 0x80101201 | Could not obtain sufficient calculation information |
| SCE_LOCATION_INFO_GET_LOCATION_CANCELED | 0x80101202 | `sceLocationGetLocation()` was cancelled |
| SCE_LOCATION_INFO_DENIED_BY_USER | 0x80101203 | Location information usage is not allowed by the user |
| SCE_LOCATION_INFO_UNDETERMINED_HEADING | 0x80101204 | Could not get direction information |
| SCE_LOCATION_INFO_DISABLE_DEVICE | 0x80101205 | Location calculating device is off and location cannot be calculated |
| SCE_LOCATION_INFO_NOT_FOUND_AP | 0x80101206 | Access points could not be found. Wi-Fi access point information will not be stored |
| SCE_LOCATION_INFO_INVALID_TIMESTAMP | 0x80101207 | Specified *timestamp* is not registered in the stored Wi-Fi access points |
| SCE_LOCATION_INFO_NETWORK_NOT_ACCESSED | 0x80101208 | Calculation will not be carried out because there is no network connection |
| SCE_LOCATION_INFO_AP_NOT_IN_LOCATION_DB | 0x80101209 | Specified access point is not stored in location database |
| SCE_LOCATION_INFO_TIMEOUT | 0x8010120A | Calculation incomplete due to timeout |

### General error

The following error codes indicate errors that can be resolved by application developer.

| Value | (Number) | Description |
|---|---|---|
| SCE_LOCATION_ERROR_INVALID_ADDRESS | 0x80101240 | Invalid data storing address |
| SCE_LOCATION_ERROR_INVALID_HANDLE | 0x80101241 | Invalid handle |
| SCE_LOCATION_ERROR_NO_MEMORY | 0x80101242 | Failed to allocate memory |
| SCE_LOCATION_ERROR_TOO_MANY_HANDLES | 0x80101243 | Could not be used because in use by many programs |

| Value | (Number) | Description |
|---|---|---|
| SCE_LOCATION_ERROR_INVALID_LOCATION_METHOD | 0x80101244 | Specified location calculation method cannot be executed |
| SCE_LOCATION_ERROR_INVALID_HEADING_METHOD | 0x80101245 | Specified direction calculation method cannot be executed |
| SCE_LOCATION_ERROR_MULTIPLE_CALLBACK | 0x80101246 | Callback already started |
| SCE_LOCATION_ERROR_NOT_RUNNING_CALLBACK | 0x80101247 | Callback has not started |
| SCE_LOCATION_ERROR_DIALOG_RESULT_NONE | 0x80101248 | Could not get dialog result |
| SCE_LOCATION_ERROR_DISABLE_APPLICATION | 0x80101249 | Application cannot perform calculation |
| SCE_LOCATION_ERROR_MULTIPLE_CONFIRM | 0x8010124A | Location information usage permission API already running |
| SCE_LOCATION_ERROR_DISABLE_DEBUG_API | 0x8010124B | Debug APIs cannot be used |
| SCE_LOCATION_ERROR_INVALID_TIMING | 0x8010124C | Invalid execution timing |
| SCE_LOCATION_ERROR_INVALID_VALUE | 0x8010124D | An invalid value was entered |
| SCE_LOCATION_ERROR_INITIALIZED | 0x8010124E | liblocation has been initialized |
| SCE_LOCATION_ERROR_UNINITIALIZED | 0x8010124F | liblocation has not been initialized |
| SCE_LOCATION_ERROR_MULTIPLE_GET_LOCATION | 0x80101250 | `sceLocationGetLocation()` or `sceLocationGetLocationWithTimeout()` is already being executed |

**Fatal errors**

The following error codes indicate fatal errors. In the case of such an error, the system developer must be contacted.

| Value | (Number) | Description |
|---|---|---|
| SCE_LOCATION_ERROR_UNAUTHORIZED | 0x80101280 | Unauthorized device |
| SCE_LOCATION_ERROR_PROVIDER_UNAVAILABLE | 0x80101281 | Could not access database |
| SCE_LOCATION_ERROR_FILE_IO | 0x80101282 | File I/O error |
| SCE_LOCATION_ERROR_INVALID_FILE_FORMAT | 0x80101283 | File format error |
| SCE_LOCATION_ERROR_TIME_OUT | 0x80101284 | Internal processing timeout error |
| SCE_LOCATION_ERROR_NO_SERVER_MEMORY | 0x80101285 | Failed to allocate server memory |
| SCE_LOCATION_ERROR_INVALID_TITLE_ID | 0x80101286 | Invalid title ID |
| SCE_LOCATION_ERROR_INTERNAL_FILE_INVALID | 0x80101287 | Invalid internal file |
| SCE_LOCATION_ERROR_FATAL_ERROR | 0x801012FF | Other fatal error |

**Debug errors**

This error returns when executing debug APIs such as GPS emulation.

| Value | (Number) | Description |
|---|---|---|
| SCE_LOCATION_DBGERROR_EMULATION_FILE_ERROR | 0x801012E0 | Specified file for GPS emulation not found |