

© 2014 Sony Computer Entertainment Inc. All Rights Reserved. SCE Confidential

# **Table of Contents**

Initialization/Termination	
SceTeleportClientInitializeInfo	
sceTeleportClientInitialize	
sceTeleportClientFinalize	
Information Obtainment	{
SceTeleportDeviceInfo	
SceTeleportDeviceDescription	10
SceTeleportRemoteAppInfo	
SceTeleportRemoteAppInfoNum	
SceTeleportCacheInfo	
SceTeleportClientGetDeviceInfoCallback	14
sceTeleportClientRegisterGetDeviceInfoCallback	15
sceTeleportClientStartSearchDevice	17
sceTeleportClientEndSearchDevice	
sceTeleportClientGetDeviceDescription	
sceTeleportClientGetRemoteAppInfoNum	2 <sup>2</sup>
sceTeleportClientGetRemoteAppInfo	23
sceTeleportClientGetCacheInfo	24
sceTeleportClientClearCacheInfo	25
Initialization/Termination of a Teleport Server Supporting Application	26
SceTeleportRemoteAppArg	
SceTeleportLaunchResponse	28
SceTeleportWakeupInfo	29
sceTeleportClientLaunchRemoteApp	
sceTeleportClientDeleteRemoteApp	33
sceTeleportClientWakeupLatestDevice	35
Teleport Server Module APIs	36
SceTeleportServerInitialInfo	
sceTeleportServerSendInitialInfo	
Common Constants	
Character String Size	
Return Codes	Δ:



# SceTeleportClientInitializeInfo

Structure storing information for initializing the teleport client module

#### **Definition**

#### **Members**

\_\_malloc Function pointer to the memory allocating function Function pointer to the memory freeing function Reserved area (fill with 0s)

### **Description**

This is a structure storing parameters to be specified to sceTeleportClientInitialize().

 $\_malloc$  is a function pointer to the memory allocating function. A function equivalent to malloc() of the C library must be passed.

\_free is a function pointer to the memory freeing function. A function equivalent to free () of the C library must be passed.

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

#### See Also

sceTeleportClientInitialize()



# sceTeleportClientInitialize

Initialize the teleport client module

#### **Definition**

### **Arguments**

info Pointer to the structure storing information for initialization

#### **Return Values**

Returns SCE OK (0) for normal termination.

Returns an error code (negative value) described in the "Return Codes" section for errors.

For a platform on which the teleport client module cannot be executed, this function returns SCE\_TELEPORT\_ERROR\_NOT\_SUPPORTED\_PLATFORM.

## **Description**

This function initializes the teleport client module. All APIs of the teleport client module must first call this function before they can be used.

For *info*, pass the information required for initializing the module. Regarding content, refer to the description of SceTeleportClientInitializeInfo.

#### **Examples**

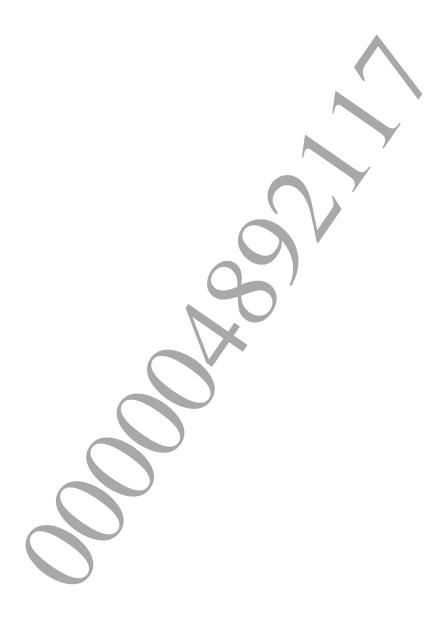
```
#include <Libsysmodule.h>
#include <teleport client.h>
returnCode = sceSysmoduleLoadModule(SCE SYSMODULE TELEPORT CLIENT);
if (returnCode != SCE_OK )
        return //
                   Error handling
SceTeleportClientInitializeInfo info;
sceClibMemset (&info,
                       0, sizeof(info));
info.malloc = malloc;
info.free = free;
returnCode = sceTeleportClientInitialize(&info);
if (returnCode != SCE OK) {
     return // Error handling
// Other sceTeleportClientXxx() call
sceTeleportClientFinalize();
sceSysmoduleUnloadModule(SCE SYSMODULE TELEPORT CLIENT);
```

Notes

This function is not multi-thread safe.

See Also

SceTeleportClientInitializeInfo



# sceTeleportClientFinalize

Terminate the teleport client module

#### **Definition**

## **Arguments**

None

### **Return Values**

Returns  $SCE_OK(0)$  for normal termination.

Returns an error code (negative value) described in the "Return Codes" section for errors.

## **Description**

This function terminates the teleport client module started with sceTeleportClientInitialize().

Successful call of sceTeleportClientInitialize() is a prerequisite to calling this function.

### **Examples**

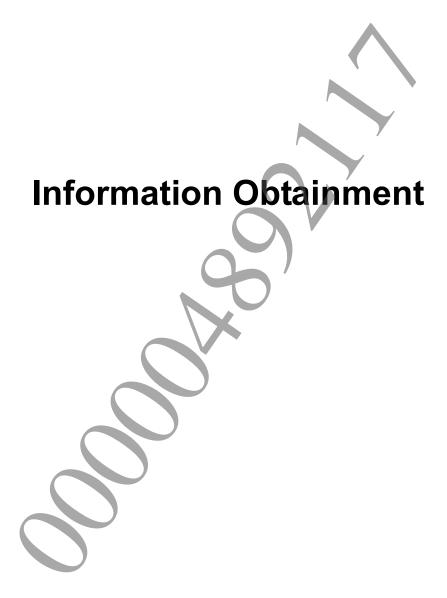
```
#include <teleport_client.h>
    :
// Initialize teleport client module
    :
sceTeleportClientFinalize ();
```

#### **Notes**

This function is not multi-thread safe.

### See Also

sceTeleportClientInitialize()



# SceTeleportDeviceInfo

Structure storing information of a teleport supporting device

#### **Definition**

#### **Members**

ipAddressinnerDatadeviceIdreservedIP address of the teleport supporting device (Ipv4)Information for internal useID for identifying the teleport supporting deviceReserved area (fill with 0s)

## **Description**

This structure stores basic information of a teleport supporting device.

innerData is a value used inside the module. Do not reference or overwrite this value.

deviceId is the ID for identifying the teleport supporting device. The value linked to the MAC address of the applicable device will be stored.

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

### See Also

sceTeleportClientRegisterGetDeviceInfoCallback(),
SceTeleportClientGetDeviceInfoCallback

# SceTeleportDeviceDescription

Structure storing supplementary information of a teleport supporting device

#### **Definition**

#### **Members**

nameName of the teleport supporting device (UTF-8)systemVersionSystem version of the teleport supporting deviceprotocolVersionVersion of teleport supported by the teleport supporting deviceinnerDataInformation for internal usereservedReserved area (fill with 0s)

## **Description**

This structure stores supplementary information of a teleport supporting device.

name is the name of the teleport supporting device.

*systemVersion* is the system version of the teleport supporting device. For the version number, bit 24 to bit 31 represents the major version and bit 12 to bit 23 represents the minor version. For example, 02.500 is represented as 0x025000000.

protocolVersion is the version of the teleport that is supported by the teleport supporting device. For the version number, bit 24 to bit 31 represents the major version and bit 12 to bit 23 represents the minor version. For example, 02.500 is represented as 0x025000000.

innerData is a value used inside the module. Do not reference or overwrite this value.

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

#### See Also

sceTeleportClientGetDeviceDescription()

# SceTeleportRemoteAppInfo

Structure storing information of a teleport server supporting application

#### **Definition**

#### **Members**

name Name of teleport server supporting application (UTF-8)

appId ID for uniquely identifying the teleport server supporting application

application application Version of the teleport server supporting application

reserved Reserved area (fill with 0s)

## **Description**

This structure stores supplementary information of a teleport server supporting application.

appVersion is the version information of the teleport server supporting application. For the version number, bit 24 to bit 31 represents the major version and bit 12 to bit 23 represents the minor version. For example, 02.500 is represented as 0x025000000.

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

#### See Also

sceTeleportClientGetRemoteAppInfo()



# **SceTeleportRemoteAppInfoNum**

Structure storing the number of teleport server supporting applications

#### **Definition**

#### **Members**

Number of teleport server supporting applications reserved Reserved area (fill with 0s)

## **Description**

This structure stores the number of teleport server supporting applications that are installed to a teleport supporting device.

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

#### See Also

sceTeleportClientGetRemoteAppInfoNum()

## SceTeleportCacheInfo

Structure storing cached information of a teleport supporting device

#### **Definition**

#### **Members**

deviceName Name of the teleport supporting device

passCode Pass code

initialTarget Whether the teleport server supporting application was terminated after it was

previously booted

deviceId ID for uniquely identifying the teleport supporting device

reserved Reserved area (fill with 0s)

## **Description**

This structure stores information that is cached by the teleport client module when booting a teleport server supporting application on a teleport supporting device using

sceTeleportClientLaunchRemoteApp() so that the next boot of the teleport server supporting application can be carried out with ease.

Information on whether the previous boot of the teleport server supporting application was terminated using sceTeleportClientDeleteRemoteApp() will be stored in <code>initialTarget</code>.

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

### See Also

sceTeleportClientClearCacheInfo()

# SceTeleportClientGetDeviceInfoCallback

Callback function to obtain device information

#### **Definition**

### **Arguments**

info Device information structure

#### **Return Values**

None

### **Description**

This callback function is specified to sceTeleportClientRegisterGetDeviceInfoCallback(). When calling sceTeleportClientStartSearchDevice() after registering the callback function, this function will be called whenever a PlayStation®TV is detected on the local area network until sceTeleportClientEndSearchDevice() is called.

Do not call APIs of this library from within the callback function. In addition, make sure processing is not stopped within the callback function.

## See Also

sceTeleportClientRegisterGetDeviceInfoCallback(),
sceTeleportClientStartSearchDevice(),sceTeleportClientEndSearchDevice()



# sceTeleportClientRegisterGetDeviceInfoCallback

Register the callback function for obtaining information of a teleport supporting device

#### **Definition**

### **Arguments**

func Pointer to the callback function

#### **Return Values**

Returns SCE OK(0) for normal termination.

Returns an error code (negative value) described in the "Return Codes" section for errors.

## **Description**

This function registers the callback function for obtaining information of a PlayStation®TV on the local area network.

Successful call of sceTeleportClientInitialize() is a prerequisite to calling this function.

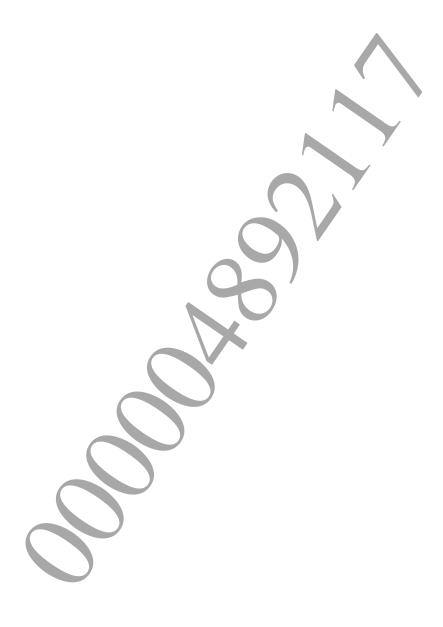
## **Examples**

#### **Notes**

This function is not multi-thread safe.

## See Also

 ${\tt sceTeleportClientInitialize(), SceTeleportClientGetDeviceInfoCallback, sceTeleportClientStartSearchDevice()}\\$ 



# sceTeleportClientStartSearchDevice

Start search of a teleport supporting device

#### **Definition**

### **Arguments**

None

#### **Return Values**

Returns SCE OK (0) for normal termination.

Returns an error code (negative value) described in the "Return Codes" section for errors.

Successful call of sceTeleportClientInitialize() is a prerequisite to calling this function.

## **Description**

This function starts the search for PlayStation®TV on the local area network with the teleport client module. When PlayStation®TV is detected as a result of this search, the callback function specified in SceTeleportClientGetDeviceInfoCallback will be called.

After being called, this function immediately returns control to the calling source.

#### **Examples**

 $Refer \ to \ the \ example \ for \ \texttt{sceTeleportClientRegisterGetDeviceInfoCallback()}.$ 

## **Notes**

This function is not multi-thread safe

### See Also

sceTeleportClientInitialize(),SceTeleportClientGetDeviceInfoCallback

# sceTeleportClientEndSearchDevice

End search of a teleport supporting device

#### **Definition**

### **Arguments**

None

#### **Return Values**

Returns SCE OK (0) for normal termination.

Returns an error code (negative value) described in the "Return Codes" section for errors.

### **Description**

This function ends the search for PlayStation®TV started with sceTeleportClientStartSearchDevice().

Successful call of sceTeleportClientInitialize() is a prerequisite to calling this function. The call of sceTeleportClientStartSearchDevice() must also have been successful.

### **Examples**

```
#include <teleport_client.h>
          :
// Initialize the teleport client module
          :
// Start device search
          :
sceTeleportClientEndSearchDevice();
```

### **Notes**

This function is not multi-thread safe.

### See Also

sceTeleportClientInitialize(), sceTeleportClientStartSearchDevice()

# sceTeleportClientGetDeviceDescription

Get supplementary information of a teleport supporting device

#### **Definition**

## **Arguments**

device
desc
desc
timeout
Waits without timing out when 0 is specified
Specify device for which supplementary information is required
Pointer to structure to store supplementary information
Timeout value (microseconds) (< INT\_MAX)
Waits without timing out when 0 is specified</pre>

### **Return Values**

Returns SCE OK (0) for normal termination.

Returns an error code (negative value) described in the "Return Codes" section for errors.

## **Description**

This function obtains supplementary information of a PlayStation®TV on the local area network. Function call is blocked until information is obtained. The timeout of the block can be specified with <code>timeout</code>.

Successful call of sceTeleportClientInitialize() is a prerequisite to calling this function. In addition, information of PlayStation®TV must be obtainable with, for example, sceTeleportClientStartSearchDevice().

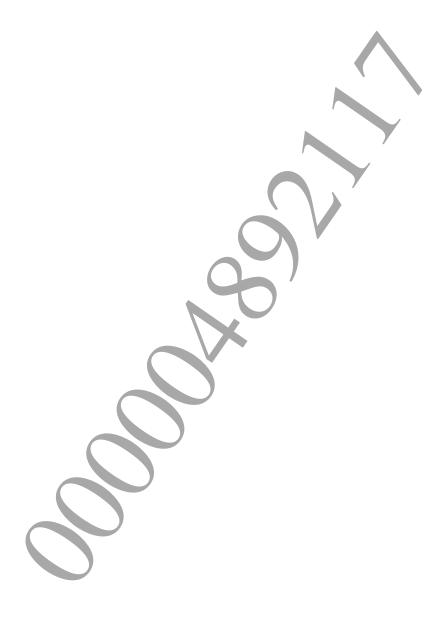
## **Examples**

## **Notes**

This function is not multi-thread safe.

## See Also

sceTeleportClientInitialize(), SceTeleportClientGetDeviceInfoCallback, sceTeleportClientStartSearchDevice(), SceTeleportDeviceInfo, SceTeleportDeviceDescription



## sceTeleportClientGetRemoteAppInfoNum

Get the number of teleport server supporting applications

#### **Definition**

## **Arguments**

device Specify device for which supplementary information is required infoNum Pointer to structure to store number of applications

Timeout value (microseconds) (< INT\_MAX)

Waits without timing out when 0 is specified

### **Return Values**

Returns SCE OK (0) for normal termination.

Returns an error code (negative value) described in the "Return Codes" section for errors.

## **Description**

This function obtains the number of teleport server supporting applications installed to a PlayStation®TV on the local area network. The function call is blocked until information is obtained. The timeout of the block can be specified with timeout.

Successful call of sceTeleportClientInitialize() is a prerequisite to calling this function. In addition, information of PlayStation®TV must be obtainable with, for example, sceTeleportClientStartSearchDevice().

## **Examples**

**©SCEI** 

## **Notes**

This function is not multi-thread safe.

### See Also

sceTeleportClientInitialize(), SceTeleportClientGetDeviceInfoCallback, sceTeleportClientStartSearchDevice(), SceTeleportDeviceInfo, SceTeleportDeviceDescription, sceTeleportClientGetRemoteAppInfo()



# sceTeleportClientGetRemoteAppInfo

Get supplementary information of a teleport server supporting application

#### **Definition**

### **Arguments**

device Specify device for which supplementary information is required size Length of \*info arrays

info Pointer to store supplementary application information timeout Timeout value (microseconds) (< INT\_MAX)

Waits without timing out when 0 is specified

#### **Return Values**

Returns SCE OK(0) for normal termination.

Returns an error code (negative value) described in the "Return Codes" section for errors.

## **Description**

This function obtains supplementary information of teleport server supporting applications that are installed to a PlayStation®TV on the local area network; the function call is blocking until information is obtained. The timeout of the block can be specified with timeout.

Arrays for the number of applications obtained with

sceTeleportClientGetRemoteAppInfoNum() must be allocated and passed to info.

Successful call of sceTeleportClientInitialize() is a prerequisite to calling this function. In addition, information of PlayStation®TV must be obtainable with, for example, sceTeleportClientStartSearchDevice().

#### **Examples**

Refer to the example for sceTeleportClientGetRemoteAppInfoNum().

#### **Notes**

This function is not multi-thread safe.

#### See Also

```
sceTeleportClientInitialize(), SceTeleportClientGetDeviceInfoCallback,
sceTeleportClientStartSearchDevice(), SceTeleportDeviceInfo,
SceTeleportDeviceDescription, sceTeleportClientGetRemoteAppInfoNum()
```

# sceTeleportClientGetCacheInfo

Get cached information of a teleport supporting device

### **Definition**

### **Arguments**

cacheInfo Cached information

#### **Return Values**

Returns SCE OK(0) for normal termination.

Returns an error code (negative value) described in the "Return Codes" section for errors.

## **Description**

This function obtains information of PlayStation®TV on which the boot of a teleport server supporting application was last successful with the call of sceTeleportClientLaunchRemoteApp().

Successful call of sceTeleportClientInitialize() is a prerequisite to calling this function.

## **Examples**

## **Notes**

This function is not multi-thread safe.

#### See Also

sceTeleportClientInitialize(), sceTeleportClientLaunchRemoteApp(),
sceTeleportClientClearCacheInfo()

# sceTeleportClientClearCacheInfo

Clear cached information of a teleport supporting device

#### **Definition**

## **Arguments**

None

#### **Return Values**

Returns SCE OK (0) for normal termination.

Returns an error code (negative value) described in the "Return Codes" section for errors.

### **Description**

This function deletes information of PlayStation®TV cached by the teleport client module.

Successful call of sceTeleportClientInitialize() is a prerequisite to calling this function.

## **Examples**

```
#include <teleport_client.h>
    :
// Initialize the teleport client module
    :
sceTeleportClientClearCacheInfo();
```

#### **Notes**

This function is not multi-thread safe

#### See Also

sceTeleportClientInitialize(), sceTeleportClientLaunchRemoteApp(), sceTeleportClientGetCacheInfo()



## **SceTeleportRemoteAppArg**

Structure storing boot argument information of a teleport server supporting application

#### **Definition**

#### **Members**

argBoot argument to provide to a teleport server supporting applicationpassCodeFour-digit pass code required to boot a teleport server supporting applicationoptionOption information to pass upon bootreservedReserved area (fill with 0s)

## **Description**

This structure stores boot argument information to pass to a teleport server supporting application upon its boot.

Characters that can be used for arg exclude 0x00 to 0x1F, 0x7F, and 0xC280 to 0xC29F in UTF-8. Specification is possible as a string including the NULL terminator that is within SCE TELEPORT LAUNCH ARG SIZE MAX bytes.

For option, specify one of the following values.

Value (Number	) Description
SCE_TELEPORT_LAUNCH_OPTION_ 0	Return an error when another application is
DEFAULT	running on the specified teleport supporting device
	and the teleport server supporting application
	cannot be booted
SCE_TELEPORT_LAUNCH_OPTION_ 1	When another application is running on the
FORCE_LAUNCH	specified teleport supporting device and the
	teleport server supporting application cannot be
	booted, terminate that running application and
	force-boot the teleport server supporting
	application

For passCode, specify the pass code that is specified to the connection destination teleport supporting device as a string between "0000" and "9999".

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

## See Also

sceTeleportClientLaunchRemoteApp()

# SceTeleportLaunchResponse

Structure storing boot result information of a teleport server supporting application

#### **Definition**

#### **Members**

appErrorHint	Stores error related information when the teleport server supporting application
	could not be booted
status	Status information returned by the teleport server supporting application
extData	Stores data specified by the teleport server supporting application
extDataBufSize	Buffer size of extData prepared by the calling source
appPort	Stores the port number that can be shared between applications when boot of the
	teleport server supporting application succeeds
reserved	Reserved area (fill with 0s)

## **Description**

This structure stores the result of the attempted boot of the teleport server supporting application.

One of the following values will be passed to status.

When used in sceTeleportServerSendInitialInfo():

Value	(Number)	Description
SCE_TELEPORT_SERVER_STATUS_DEFAULT	0	Teleport server supporting application
		accepted connection
SCE_TELEPORT_SERVER_STATUS_REFUSED	1	Teleport server supporting application
		rejected connection

When used in sceTeleportClientLaunchRemoteApp():

Value	(Number)	Description
SCE_TELEPORT_CLIENT_RESPONSE_STATUS_	0	Teleport server supporting application
DEFAULT		was successfully booted
SCE_TELEPORT_CLIENT_RESPONSE_STATUS_	1	Teleport server supporting application
ALREADY_LAUNCHED		was already booted

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

## See Also

sceTeleportClientGetRemoteAppInfoNum()

# **SceTeleportWakeupInfo**

Structure storing information for booting a teleport supporting device

### **Definition**

### **Members**

reserved Reserved area (fill with 0s)

### **Description**

This structure is passed to the sceTeleportClientWakeupLatestDevice() function that boots a teleport supporting device via network.

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

## See Also

sceTeleportClientWakeupLatestDevice()

# sceTeleportClientLaunchRemoteApp

Boot a teleport server supporting application

#### **Definition**

## **Arguments**

#### **Return Values**

Returns SCE OK (0) for normal termination.

Returns an error code (negative value) described in the "Return Codes" section for errors.

## **Description**

This function boots the teleport server supporting application that is installed to a PlayStation®TV on the local area network; the function call is blocking until the application is booted. The timeout of the block can be specified with <code>timeout</code>.

For appInfo, specify the application information that can be obtained with sceTeleportClientGetRemoteAppInfo().

When application boot succeeds or when the target application is already running and the application successfully calls <code>sceTeleportServerSendInitialInfo()</code>, the port number specified by the teleport server supporting application will be stored in <code>res.appPort</code>. The teleport server supporting application and teleport client supporting application can communicate using this port.

If another application on the connection destination PlayStation®TV is blocking the boot of the teleport server supporting application, the name of the blocking application will be encoded in UTF-8 and stored in res.appErrorHint. The application that called this function can use this information to prompt the user to terminate the application that is running on the connection destination PlayStation®TV.

When the teleport server supporting application rejects connection,

 ${\tt SCE\_TELEPORT\_SERVER\_STATUS\_REFUSED} \ will \ be \ stored \ in \ \textit{res.status}. \ If \ the \ teleport \ server \ supporting \ application \ has \ already \ been \ booted,$ 

```
SCE_TELEPORT_CLIENT_RESPONSE_STATUS_ALREADY_LAUNCHED will be stored in res.status.
```

If PlayStation®TV has a pass code specified and this function is called without a pass code specified or with the wrong pass code specified to <code>argInfo.passCode</code>, this function returns <code>SCE\_TELEPORT\_ERROR\_LAUNCH\_FAILED\_INVALID\_PASSCODE</code>. Handle this error and prompt the user to enter the pass code.

**©SCEI** 

When boot of the teleport server supporting application succeeds, the client module caches information equivalent to SceTeleportCacheInfo of the connection destination PlayStation®TV. This information can be obtained with sceTeleportClientGetCacheInfo().

Successful call of sceTeleportClientInitialize() is a prerequisite to calling this function. In addition, information of PlayStation®TV must be obtainable with, for example, sceTeleportClientStartSearchDevice(). Moreover, information of the boot target application must be obtainable with sceTeleportClientGetRemoteAppInfo().

### **Examples**

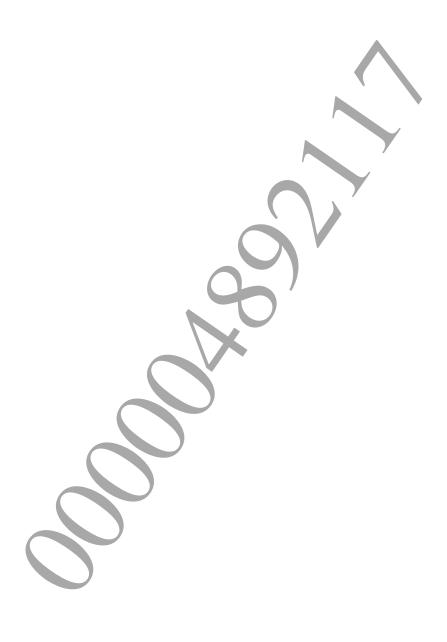
```
#include <teleport client.h>
// Initialize the teleport client module
// Obtain device information
// Obtain application information
SceTeleportLaunchResponse res;
SceTeleportRemoteAppArg contentInfo;
memset(&res, 0, sizeof(SceTeleportLaunchResponse));
memset(&contentInfo, 0, sizeof(SceTeleportRemoteAppArg));
ret = sceTeleportClientLaunchRemoteApp(
         (const SceTeleportDeviceInfo *) (&deviceInfo),
         (const SceTeleportRemoteAppInfo *)(&appInfo),
         (const SceTeleportRemoteAppArg
                                        *) &contentInfo,
        &res,
         (10*1000*1000);
if(ret == SCE TELEPORT ERROR LAUNCH FAILED INVALID PASSCODE) {
        // Enter pass code
        //contentInfo.passCode[0] = '1';
        //contentInfo.passCode[1]
        //contentInfo.passCode[2] = '3';
        //contentInfo.passCode[3] = '4';
        ret
               sceTeleportClientLaunchRemoteApp(
               (const SceTeleportDeviceInfo *)(&deviceInfo),,
               (const SceTeleportRemoteAppInfo *)(&appInfo),
               const SceTeleportRemoteAppArg *)&contentInfo,
               res,
               (10*1000*1000));
}
```

## Notes

This function is not multi-thread safe.

## See Also

sceTeleportClientInitialize(), SceTeleportClientGetDeviceInfoCallback,
sceTeleportClientStartSearchDevice(), SceTeleportDeviceInfo,
SceTeleportDeviceDescription, sceTeleportClientGetRemoteAppInfoNum(),
sceTeleportClientGetRemoteAppInfo(), sceTeleportClientGetCacheInfo()



# sceTeleportClientDeleteRemoteApp

Terminate a teleport server supporting application

#### **Definition**

## **Arguments**

device appInfo Information of the termination target application
timeout Timeout value (microseconds) (< INT\_MAX)
Waits without timing out when 0 is specified</pre>

### **Return Values**

Returns SCE OK (0) for normal termination.

Returns an error code (negative value) described in the "Return Codes" section for errors.

## **Description**

This function terminates the teleport server supporting application booted with sceTeleportClientLaunchRemoteApp(); the function call is blocking until the application is terminated. The timeout of the block can be specified with timeout.

For appInfo, specify the application information that can be obtained with sceTeleportClientGetRemoteAppInfo().

When sceTeleportClientLaunchRemoteApp() is called for an application that is already running, that application cannot be terminated with this function.

Successful call of sceTeleportClientInitialize() is a prerequisite to calling this function. In addition, information of PlayStation®TV must be obtainable with, for example, sceTeleportClientStartSearchDevice(). Moreover, information of the boot target application must be obtainable with sceTeleportClientGetRemoteAppInfo().

### **Examples**

#### **Notes**

This function is not multi-thread safe.

#### See Also

sceTeleportClientInitialize(), SceTeleportClientGetDeviceInfoCallback,
sceTeleportClientStartSearchDevice(), SceTeleportDeviceInfo,
SceTeleportDeviceDescription, sceTeleportClientGetRemoteAppInfoNum(),
sceTeleportClientGetRemoteAppInfo(), sceTeleportClientLaunchRemoteApp()



# sceTeleportClientWakeupLatestDevice

Issue a WakeOnLan signal to a teleport supporting device

#### **Definition**

## **Arguments**

info Supplementary information for the boot

#### **Return Values**

Returns SCE OK(0) for normal termination.

Returns an error code (negative value) described in the "Return Codes" section for errors.

## **Description**

This function issues a WakeOnLan signal to a PlayStation®TV that is cached by the teleport client module.

Successful call of sceTeleportClientInitialize() is a prerequisite to calling this function.

## **Examples**

```
#include <teleport_client.h>
    :
// Initialize the teleport client module
    :
sceTeleportClientWakeupLatestDevice();
```

#### **Notes**

This function is not multi-thread safe

#### See Also

 $\label{lem:control} sceTeleportClientInitialize(), sceTeleportClientLaunchRemoteApp(), sceTeleportClientGetCacheInfo()\\$ 



## SceTeleportServerInitialInfo

Structure storing initialization information of a teleport server supporting application

### **Definition**

#### **Members**

statusStatus informationextDataAdditional information specified by the applicationportPort number used for communication between applicationsreservedReserved area (fill with 0s)

## **Description**

This structure stores information that is sent with sceTeleportServerSendInitialInfo() upon boot of a teleport server supporting application.

For extData, a string of length SCE\_TELEPORT\_SERVER\_EXT\_DATA\_MAX\_SIZE or less (including the NULL terminator) must be specified. Characters that can be used exclude 0x00 to 0x1F, 0x7F, and 0xC280 to 0xC29F in UTF-8.

Specify one of the following values for status.

Value	(Number)	Description
SCE_TELEPORT_SERVER_STATUS_DEFAULT	0	Allows connection from a teleport client
		supporting application
SCE_TELEPORT_SERVER_STATUS_REFUSED	1	Rejects connection from a teleport client
		supporting application

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

### See Also

sceTeleportServerSendInitialInfo()

## sceTeleportServerSendInitialInfo

Send initialization information to a teleport client supporting application

#### **Definition**

## **Arguments**

#### **Return Values**

Returns SCE OK(0) for normal termination.

Returns an error code (negative value) described in the "Return Codes" section for errors.

## **Description**

This function is used by teleport server supporting applications. This function sends initialization information to a teleport client supporting application on the local area network; the function call is blocking until send is completed. The timeout of the block can be specified with timeout.

A teleport server supporting application must correctly handle being booted by a teleport client supporting application and call this function with the appropriate value set to <code>info</code>.

For the handling of a boot by the Teleport library, refer to the "Notes" chapter of the "Teleport Library Overview" document.

To reject connection from a teleport client supporting application, call this function with SCE TELEPORT SERVER STATUS REFUSED specified to *info.status*.

For *info.port*, specify the port number to use for communication with the teleport client supporting application.

### **Examples**

```
#include <teleport_server.h>

// Check boot by Teleport
:
SceTeleportServerInitialInfo initTeleportInfo;
memset(&initTeleportInfo,NULL,sizeof(initTeleportInfo));
initTeleportInfo.port = s_appCommPort;

ret = sceTeleportServerSendInitialInfo(&initTeleportInfo, 10*1000*1000);
```

#### Notes

This function is not multi-thread safe.

**©SCEI** 

## See Also

sceTeleportClientLaunchRemoteApp()





# **Character String Size**

Maximum size of character strings

## Definition

Value	(Number)	Description
SCE_TELEPORT_LAUNCH_ARG_SIZE_MAX	952	Maximum size of boot argument
		string
SCE_TELEPORT_SERVER_EXT_DATA_MAX_SIZE	128	Maximum size of additional
		information that can be sent by a
		teleport server supporting application

## **Description**

These constants represent maximum sizes for the exchange of strings in this library. The values include the NULL terminator.



# **Return Codes**

List of return codes returned by the Teleport library

## Definition

Value	(Number)	Description
SCE_TELEPORT_ERROR_FATAL	0x80108401	Fatal error
SCE_TELEPORT_ERROR_PARAM	0x80108402	Invalid parameter
SCE TELEPORT ERROR	0x80108403	Teleport client module is not initialized
NOT_INITIALIZED		•
SCE_TELEPORT_ERROR_	0x80108404	Attempted to initialize an already
ALREADY_INITIALIZED		initialized teleport client module
SCE_TELEPORT_ERROR_STATE	0x80108405	Error regarding function call order
SCE_TELEPORT_ERROR_XML_PARSE	0x80108406	Error regarding XML parse
SCE_TELEPORT_ERROR_NO_MEMORY	0x80108407	Insufficient memory
SCE_TELEPORT_ERROR_	0x80108408	Communication timed out
CONNECTION_TIMEOUT		
SCE_TELEPORT_ERROR_	0x80108409	Error regarding communication
COMMUNICATION		
SCE_TELEPORT_ERROR_	0x8010840a	Unsupported API call
NOT_SUPPORTED_PLATFORM	0.00100100	- 111
SCE_TELEPORT_ERROR_ LAUNCH FAILED INVALID PASSCODE	0x80108430	Invalid pass code
SCE TELEPORT ERROR	0x80108431	Post failed because another application is
LAUNCH FAILED OTHER APP FG	0x60106451	Boot failed because another application is
SCE TELEPORT ERROR	0x80108432	running (FG)
LAUNCH_FAILED_OTHER_APP_BG	000108432	Boot failed because another application is
	0.00100400	running (BG)
SCE_TELEPORT_ERROR_ LAUNCH FAILED APP REFUSED	0x80108433	Rejected by application
SCE_TELEPORT_ERROR_LAUNCH_FAILED	0x80108434	Failed to boot application
SCE_TELLET ONT_ERROR_LAUNCH_FATTED	000100434	(other error occurred - boot may succeed
		when it is reattempted)
SCE TELEPORT ERROR	0x80108435	± /
DELETE FAILED FORBIDDEN	UX8U1U8435	Failed to terminate application
	0. 00100407	(application termination is not permitted)
SCE_TELEPORT_ERROR_ DELETE FAILED	0x80108436	Failed to terminate application
_	0.00100407	(other error occurred)
SCE_TELEPORT_ERROR_	0x80108437	Failed to boot application
LAUNCH_FAILED_INTERNAL		(internal error occurred)
SCE_TELEPORT_ERROR	0x80108438	Failed to boot application (system update
LAUNCH_FAILED_VITATV_NEED_UPDATE		required on PlayStation®TV)