Teleport Library Overview

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Table of Contents

1 Library Overview

Teleport

"Teleport" is a network protocol for controlling boot/termination of applications running on other devices from devices in the same local area network.

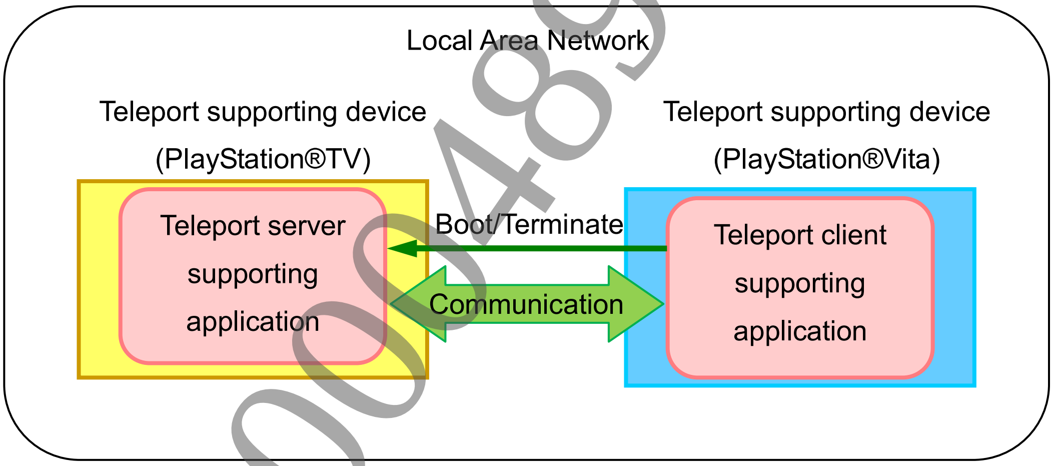
From among teleport supporting applications, the applications with teleport client features are called "teleport client supporting applications", and applications with teleport server features are called "teleport server supporting applications". A teleport client supporting application running on a teleport supporting device can control the boot/termination of teleport server supporting applications running on other teleport supporting devices in the same local area network.

Note
Currently, the only device that supports teleport server features is the PlayStation®TV.

Situations such as streaming playback of video files on a PlayStation®Vita to a PlayStation®TV in order to view them on a large screen TV are possible situations where teleport features are useful. Since both of the applications running on the respective devices at this time support teleport, the booting and termination of the playback application on the PlayStation®TV can be controlled from the PlayStation®Vita.

Therefore, users will not need to control the PlayStation®TV itself in order to perform these operations.

Figure 1 Teleport Communication Diagram


Local Area Network
Teleport supporting device
(PlayStation®TV)
Teleport supporting device
(PlayStation®Vita)
Teleport server supporting application
Boot/Terminate
Communication
Teleport client supporting application

Purpose and Characteristics

The Teleport library is a collection of features required to develop teleport supporting applications that run on PlayStation®Vita/PlayStation®TV. By using the Teleport library, the teleport network protocol part of implementations will be hidden from applications, therefore developers can develop teleport supporting applications without needing to know the details of the teleport network protocol.

The Teleport library is comprised of two modules: the teleport client module and teleport server module. An application that uses the teleport server module is referred to as a teleport server supporting application. Applications will run as teleport server supporting applications by using the teleport server module, and applications will run as teleport client supporting applications by using the teleport client module.

The teleport client module is used in an application on PlayStation®Vita and the teleport server module is used in an application on PlayStation®TV. PlayStation®TV and PlayStation®Vita can execute the same binary application; thus, it is possible to have one application with features of both the teleport server module and the teleport client module with the same application running on both PlayStation®TV and PlayStation®Vita.

Main Features

Teleport Server Module Features

The teleport server module provides the following features.

- Responding to boot/termination requests from teleport client supporting applications running on teleport supporting devices in local area networks
- Sending information to teleport client supporting applications running on teleport supporting devices in local area networks

Teleport Client Module Features

The teleport client module provides the following features.

- Detecting teleport supporting devices in local area networks
- Obtaining information on teleport server supporting applications running on teleport supporting devices in local area networks
- Booting/terminating teleport server supporting applications running on teleport supporting devices in local area networks
- Issuing WakeOnLan signals to teleport supporting devices in local area networks

Used Resources

The Teleport library uses the following system resources.

Teleport Client Module (SCE_SYSMODULE_TELEPORT_CLIENT)

Resource	Description
Footprint	68 KiB
Work memory	16 KiB from the memory allocator provided by the application
Thread	Creates one thread each upon the call of `sceTeleportClientStartSearchDevice()` and `sceTeleportClientWakeupLatestDevice()`. A maximum of three threads can be running at the same time. The stack size of one thread is 4 KiB.

Teleport Server Module (SCE_SYSMODULE_TELEPORT_SERVER)

Resource	Description
Footprint	8 KiB
Work memory	None used in particular
Thread	Not internally created

Embedding into a Program

The following files are required in order to use the Teleport library.

Teleport Client Module

Filename	Description
teleport_client.h	Header file
libSceTeleportClient_stub.a libSceTeleportClient_stub_weak.a	Stub library file

Teleport Server Module

Filename	Description
teleport_server.h	Header file
libSceTeleportServer_stub.a libSceTeleportServer_stub_weak.a	Stub library file

If the server and client applications differ, include teleport_server.h in the application on PlayStation®TV. Include teleport_client.h in the application on PlayStation®Vita.

If the server application is the same as the client application, include both teleport_server.h and teleport_client.h in the source program (in addition, several header files will automatically be included).

libnet must be initialized before using the client module of the Teleport library. For instructions on initialization, refer to the "libnet Overview" document.

Load the required PRX modules in the program to use the Teleport library, as follows.

```
// Teleport client module
returnCode = sceSysmoduleLoadModule(SCE_SYSMODULE_TELEPORT_CLIENT);
if (returnCode != SCE_OK ) {
        return // Error handling
}

// Teleport server module
returnCode = sceSysmoduleLoadModule(SCE_SYSMODULE_TELEPORT_SERVER);
if (returnCode != SCE_OK ) {
        return // Error handling
}
```

To use the Teleport library, statically link libSceTeleportClient_stub.a or libSceTeleportServer_stub.a (or both).

The application utility must be used for the development of a teleport supporting application. For details, refer to the "How to Confirm Whether the Boot Is by the Teleport Library" section of Chapter 3 "Notes".

Sample Program

The following sample program uses the Teleport library.

- sample_code/network/api_teleport/

2 Using the Library

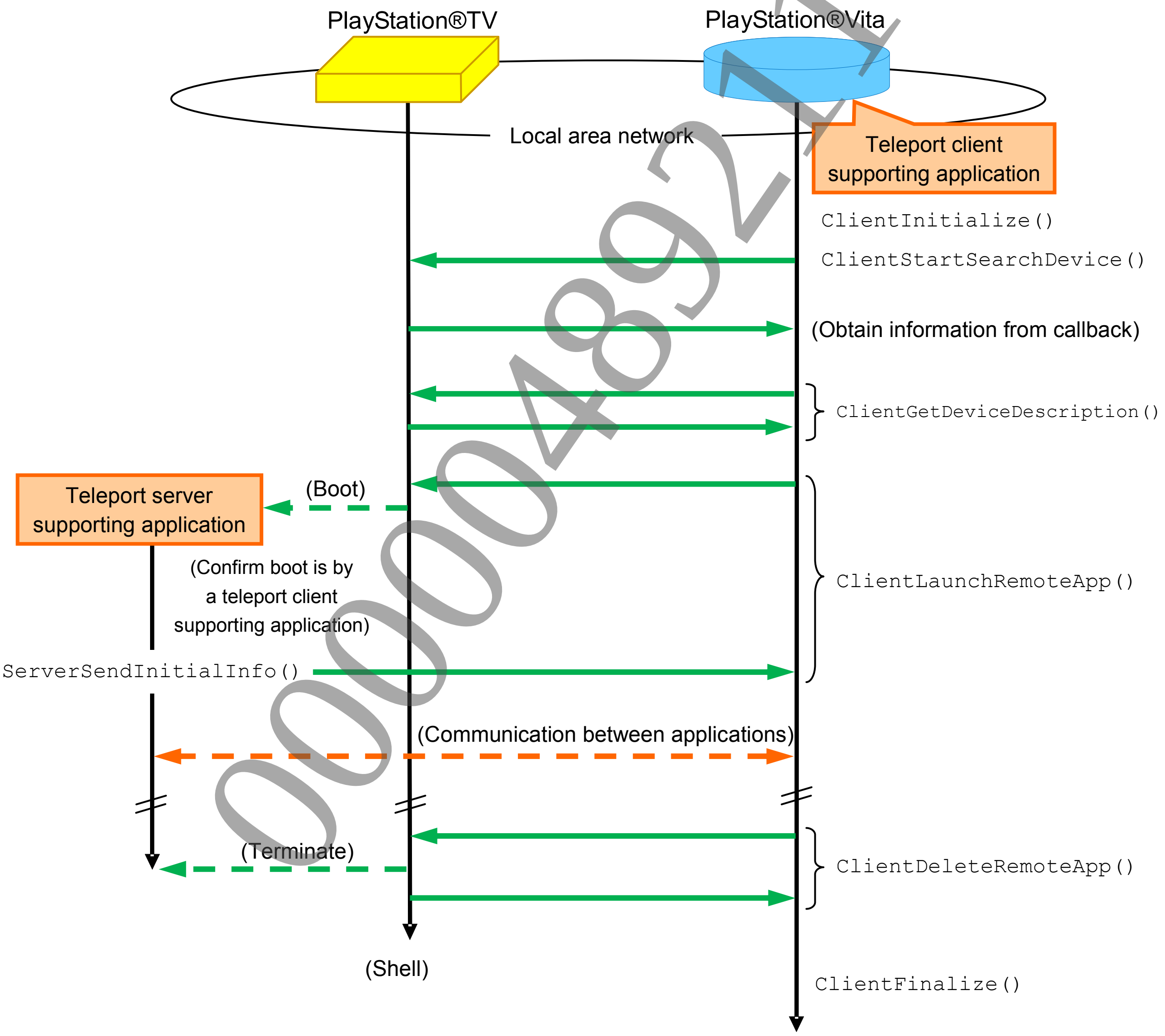
Overview

Communication between PlayStation®Vita and PlayStation®TV is indicated in the figure below. The teleport client supporting application on PlayStation®Vita calls APIs of the Teleport library to boot, terminate, and obtain information of, the Teleport server supporting application on PlayStation®TV.

Processing will be blocked while `sceTeleportClientGetDeviceDescription()`, `sceTeleportClientLaunchRemoteApp()`, or `sceTeleportClientDeleteRemoteApp()` is called.

For the procedure on how to detect that a teleport server supporting application was booted from a teleport client supporting application, refer to Chapter 3 "Notes".

Figure 2 Processing Procedure


PlayStation®TV
PlayStation®Vita
Local area network
Teleport client supporting application
ClientInitialize()
ClientStartSearchDevice()
(Obtain information from callback)
ClientGetDeviceDescription()
Teleport server supporting application
(Boot)
(Confirm boot is by a teleport client supporting application)
ClientLaunchRemoteApp()
ServerSendInitialInfo()
(Communication between applications)
(Terminate)
ClientDeleteRemoteApp()
(Shell)
ClientFinalize()

*The `sceTeleport` function prefix is omitted.

Processing Procedure of the Teleport Server Supporting Application

(1) Confirm whether the boot is by the Teleport library

(2) Send initialization information to the teleport client supporting application

(1) Confirm whether the boot is by the Teleport library

Upon boot, the teleport server supporting application must evaluate whether or not it has been started by the Teleport library. For instructions, refer to the "How to Confirm Whether the Boot Is by the Teleport Library" section of Chapter 3 "Notes".

(2) Send initialization information to the teleport client supporting application

After confirming the boot is by the Teleport library, call `sceTeleportServerSendInitialInfo()` to send initialization information to the teleport client supporting application.

Processing Procedure of the Teleport Client Supporting Application

(1) Initialize the teleport client module

(2) Search for PlayStation®TV

(3) Obtain information of the teleport server supporting application

(4) Boot a teleport server supporting application

(5) Terminate the teleport server supporting application

(6) Terminate the teleport client module

(1) Initialize the teleport client module

Call `sceTeleportClientInitialize()` to initialize the teleport client module.

(2) Search for PlayStation®TV

Call `sceTeleportClientRegisterGetDeviceInfoCallback()` to set a callback function and call `sceTeleportClientStartSearchDevice()`. This callback function will be called every time a PlayStation®TV is subsequently detected on the local network.

As input to the callback function, the `SceTeleportDeviceInfo` structure will be passed for identifying the PlayStation®TV.

(3) Obtain information of the teleport server supporting application

Use the structure for identifying PlayStation®TV obtained in step (2) and call `sceTeleportClientGetRemoteAppInfoNum()` and `sceTeleportClientGetRemoteAppInfo()` to, respectively, obtain the number of teleport server supporting applications installed to PlayStation®TV and the information of the teleport server supporting applications.

(4) Boot a teleport server supporting application

Use the structure for identifying PlayStation®TV obtained in step (2) and information of teleport server supporting applications obtained in step (3) and call `sceTeleportClientLaunchRemoteApp()` to boot the applicable teleport server supporting application.

(5) Terminate the teleport server supporting application

Use the structure for identifying PlayStation®TV obtained in step (2) and information of teleport server supporting applications obtained in step (3) and call `sceTeleportClientDeleteRemoteApp()` to terminate the booted teleport server supporting application.

(6) Terminate the teleport client module

Call `sceTeleportClientFinalize()` to carry out termination processing for the teleport client module.

Issuing a WakeOnLan Signal to a Teleport Supporting Device

When `sceTeleportClientWakeupLatestDevice()` is called by a teleport client supporting application, a WakeOnLan signal is issued to the PlayStation®TV that is being cached by the teleport client module.

Internally, `sceTeleportClientStartSearchDevice()` is called with the call of `sceTeleportClientWakeupLatestDevice()` and a search is made for a teleport supporting device.

By receiving the response returned from the target PlayStation®TV with `SceTeleportClientGetDeviceInfoCallback`, the teleport client supporting application can detect the wakeup of PlayStation®TV. In this wakeup state, PlayStation®TV does not yet output to the connected display. The detection of PlayStation®TV wakeup triggers the call of `sceTeleportClientLaunchRemoteApp()` by the teleport client supporting application to boot a teleport server supporting application, and this will prompt the target PlayStation®TV to start outputting to the connected display.

List of Functions

The list of functions provided by the Teleport library is as follows.

Teleport Client Module

Function	Description
`sceTeleportClientInitialize()`	Initializes the teleport client module
`sceTeleportClient RegisterGetDeviceInfoCallback()`	Registers the callback function for obtaining information of a teleport supporting device
`sceTeleportClientStartSearchDevice()`	Starts search of a teleport supporting device
`sceTeleportClientGetDeviceDescription()`	Gets supplementary information of a teleport supporting device
`sceTeleportClientGetRemoteAppInfoNum()`	Gets the number of teleport server supporting applications
`sceTeleportClientGetRemoteAppInfo()`	Gets supplementary information of a teleport server supporting application
`sceTeleportClientLaunchRemoteApp()`	Boots a teleport server supporting application
`sceTeleportClientDeleteRemoteApp()`	Terminates a teleport server supporting application
`sceTeleportClientGetCacheInfo()`	Gets cached information of a teleport supporting device
`sceTeleportClientClearCacheInfo()`	Clears cached information of a teleport supporting device
`sceTeleportClientWakeupLatestDevice()`	Issues a WakeOnLan signal to a teleport supporting device
`sceTeleportClientEndSearchDevice()`	Ends search of a teleport supporting device
`sceTeleportClientFinalize()`	Terminates the teleport client module

Teleport Server Module

Function	Description
`sceTeleportServerSendInitialInfo()`	Sends initialization information to a teleport client supporting application

3 Notes

Notes for creating a teleport server supporting application are explained below.

WakeOnLan Operation Check on Development Kits

When checking WakeOnLan operation using a Development Kit with **PS TV Emulation** set to **On** as the receiving side, use a Wi-Fi device as the network device (do not use a wired USB Ethernet device).

How to Confirm Whether the Boot Is by the Teleport Library

Upon boot, a teleport server supporting application must evaluate whether or not it has been started by the Teleport library. Include apputil/apputil_launch_app.h in the application utility and call APIs as follows to make the evaluation.

```
#include <apputil.h>

// Check startup parameters
SceAppUtilAppEventParam appEventParam;
memset(&appEventParam, 0, sizeof(SceAppUtilAppEventParam));

/* Receive application event content */
ret = sceAppUtilReceiveAppEvent( &appEventParam );
if( (ret==SCE_OK) &&
    (appEventParam.type == SCE_APPUTIL_APPEVENT_TYPE_TELEPORT) )
{
    /* Boot by the Teleport library */
    SceAppUtilTeleportParam param;
    memset(&param, 0, sizeof(SceAppUtilTeleportParam));

    /* Parse application startup parameters from Teleport and store the result
*/
    ret = sceAppUtilAppEventParseTeleport( &appEventParam, &param );
}
```

param.sfo Settings

To enable boot by the Teleport library from a teleport client supporting application, parameters of param.sfo must be set as follows upon package creation of teleport server supporting applications. Make these settings with Param File Editor. For details, refer to the "Param File Editor User's Guide" document included in Publishing Tools.

- To indicate that the application is a teleport server supporting application, enable "The Application can be launched by Teleport feature"
- To enable the application to start on PlayStation®TV, enable "PS Vita:Bootable, PS TV:Bootable"