Application Manager Overview

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Table of Contents

1 Library Overview

Purpose and Features

Application manager is a kernel module that manages and adjusts the application status and distributes events. The system calls that can be called by an application are public.

Currently, the public functions are as follows.

- Function for obtaining application state
- Function for setting info bar state
- Function for setting the Network Disconnection Warning Dialog state
- Function for adjusting other applications and obtaining the BGM output rights
- Function for switching to another program

For more information on the function for obtaining the BGM output rights, refer to the "BGM Port Control System Call Overview" and "BGM Port Control System Call Reference" documents.

Application State

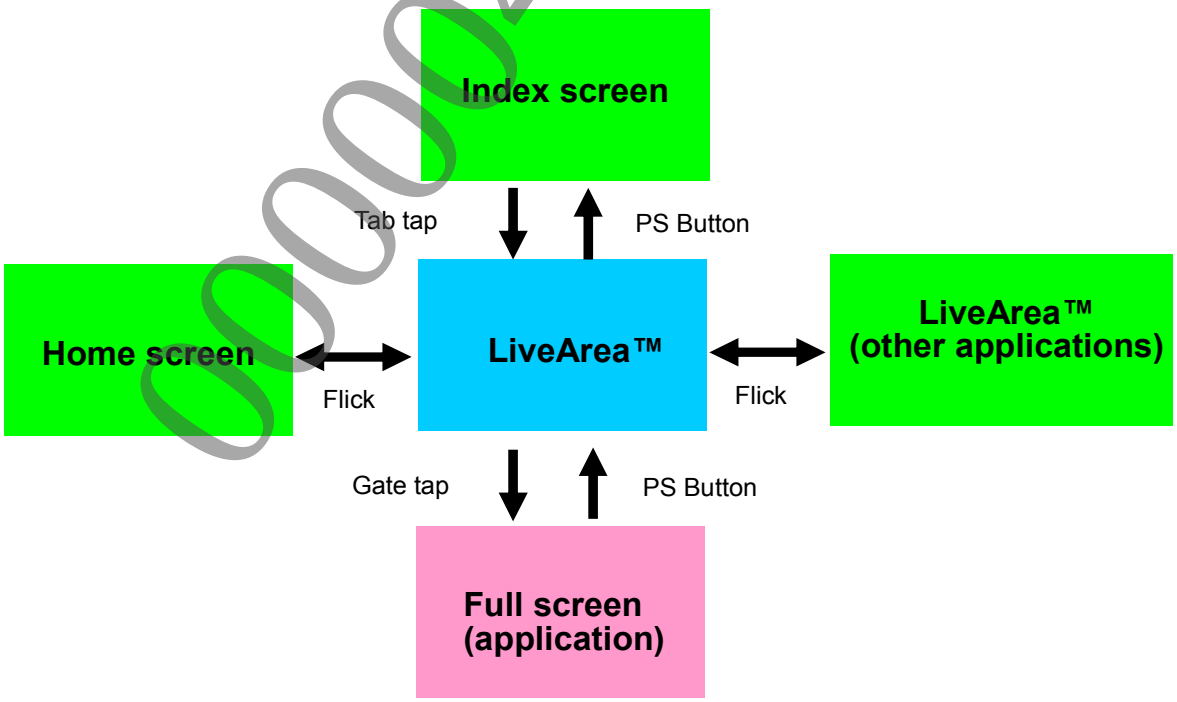
System Events

Resume Event

The screen of the system software for PlayStation®Vita changes as shown in Figure 1.

The state when an application is displayed on the screen is full screen. In other states (such as LiveArea™), an application is not displayed.

Figure 1 Screen Transition of PlayStation®Vita



In addition, when the state is full screen, the application starts or resumes, and when the state is "LiveArea™", the application is suspended.

Although the application itself cannot know whether it is suspended, the application can know that it has resumed when a system event of the application manager is retrieved.

There are no processes required on the system when resuming. Use the events according to the application specifications, such as to temporarily stop the application.

NP Message Event

This event notifies the application that an NP message has arrived. This system event overwrites without queuing, so only one event is delivered even if multiple NP messages arrive until the application retrieves an event.

For how to perform processing when receiving this event, refer to the "NP Message Overview" document.

Store Checkout Event

There are cases where you may want to have the application monitor purchases made by the user at the PlayStation®Store and perform a certain action at the time of a certain purchase. For this reason, each time a user makes a purchase at the PlayStation®Store, a store checkout event is distributed to the application that is running.

This system event overwrites without queuing, so only one event is delivered even if multiple store checkout events arrive until the application retrieves an event.

The interface for processing a store checkout event is expected to be added in a future release.

Store Redemption Event

This event notifies the application that a promotion code was redeemed at the PlayStation®Store. This system event overwrites without queuing, so only one event is delivered even if multiple store redemption events arrive until the application retrieves an event.

Application Event

For how to process an application event, refer to the "Application Utility Overview" document.

UI Overlaid State of System Software

When the UI of the system software is overlaid, the application loses possession of the "Controller Service" and "Touch Service", and data input is lost.

There are no processes required on the system when resuming. Use that state information according to the application specifications, such as to temporarily stop the application.

For the behavior when the application loses possession of the "Controller Service" and "Touch Service", refer to the "Controller Service Overview" and "Touch Service Overview" documents.

Info Bar Setting

The display setting of the info bar can be changed while running an application. Note that, depending on conditions, the info bar setting specified by the system may take priority.

By default, the info bar is invisible.

Network Disconnection Warning Dialog Setting

It is possible to set the Network Disconnection Warning Dialog state when an application is running. By enabling Network Disconnection Warning Dialog, a warning dialog will be displayed when the PS button is pressed and the application is paused during usage of the network functions. Since enabling Network Disconnection Warning Dialog is a feature that is only permitted for applications that use the network functions, do not enable it for applications that do not use the network functions.

The initial state for the Network Disconnection Warning Dialog state is disabled.

Switching to Another Program

It is possible to switch from a currently running program to another specified program.

Files

Files required for using the system calls of the application manager are as follows.

Filename	Description
appmgr.h	Header file
libSceAppMgr_stub.a	Stub library file

To use these, statically link libSceAppMgr_stub.a.

Sample Programs

Refer to the following sample program for the system calls of the application manager.

samples/sample_code/system/api_appmgr/

This sample shows basic uses of the application manager.

2 Basic Procedure

Obtaining Application State

The basic procedure for obtaining the system events and the UI overlaid state of the system software as the application state with the application manager is shown below.

(1) Get number of events

The application manager has an event queue for each running application.

The `sceAppMgrGetAppState()` can be used to obtain the number of system and application events.

```
SceAppMgrSystemEvent systemEvent
SceAppMgrAppState appState;

ret = sceAppMgrGetAppState ( &appState );
```

(2) Get system event details

When there is more than one event, call the `sceAppMgrReceiveSystemEvent()` for each event to obtain the event details.

Calling this function deletes the event from the event queue.

```
if ( ( ret == SCE_OK ) && (appState.systemEventNum > 0 ) )
{
        for ( i = 0; i < appState.systemEventNum; i++ )
        {
              ret = sceAppMgrReceiveSystemEvent( &systemEvent );
```

(3) Perform processing according to system event ID

Perform processing according to the type of event, while referring to the event field of the `SceAppMgrSystemEvent` structure.

```
        ret = sceAppMgrReceiveSystemEvent( &systemEvent );
        if( systemEvent.event == SCE_APPMGR_SYSTEMEVENT_ON_RESUME)
        {
              /* Processing at Resume state */
        }
        else if( systemEvent.event ==
                    SCE_APPMGR_SYSTEMEVENT_ON_NP_MESSAGE_ARRIVED)
        {
              /* Processing when NP message event arrives */
        }
        else if( systemEvent.event ==
                    SCE_APPMGR_SYSTEMEVENT_ON_STORE_PURCHASE)
        {
              /* Processing when store checkout event arrives */
        }
}
```

(4) Perform processing according to UI overlaid state of system software

Perform processing according to the UI overlaid state of system software, while referring to the *isSystemUiOverlaid* field of the SceAppMgrAppState structure.

```
if ( ret == SCE_OK )
{
        if ( appState.isSystemUiOverlaid == SCE_TRUE )
        {
              /* Processing when UI of system software is overlaid */
        }
        else
        {
              /* Processing when UI of system software is not overlaid */
        }
}
```

Setting the Info Bar

Specify the visibility, color, and transparency of the info bar and execute sceAppMgrSetInfobarState().

```
int visibility = SCE_APPMGR_INFOBAR_VISIBILITY_VISIBLE;
int color = SCE_APPMGR_INFOBAR_COLOR_BLACK;
int transparency = SCE_APPMGR_INFOBAR_TRANSPARENCY_OPAQUE;
sceAppMgrSetInfobarState(visibility, color, transparency);
```

Setting Network Disconnection Warning Dialog

Execute sceAppMgrSetNetworkDisconnectionWarningDialogState() and set Network Disconnection Warning Dialog to enabled/disabled.

```
sceAppMgrSetNetworkDisconnectionWarningDialogState(SCE_TRUE, NULL);
```

Switching to Another Program

When sceAppMgrLoadExec() is used, the program that is currently running is terminated and another program is used to run the application again.

```
char *const argv[] = {"argument", NULL};
sceAppMgrLoadExec("app0:other.self", argv, NULL);
```

List of Functions

The system calls of the application manager are listed below.

For more information, refer to the "Application Manager Reference" document.

Functions	Description
sceAppMgrGetAppState()	Gets the application status (such as the number of system events that arrived)
sceAppMgrReceiveSystemEvent()	Gets event details
sceAppMgrSetInfobarState()	Sets the info bar state
sceAppMgrSetNetworkDisconnectionWarningDialogState()	Sets the Network Disconnection Warning Dialog state
sceAppMgrLoadExec()	Switches the application to another program