System Software Overview

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Document serial number: 000004892117

Table of Contents

1 Overview

Scope of This Document

This document explains the development support functions and the setting-related functions that have an influence on the operation of the application among the functions that are provided in the system software of the Development Kit (DevKit) and the Testing Kit (TestKit). It also describes the applications (Music, etc.) provided as the system software.

The sequence of contents covering the above-mentioned development support functions and setting-related functions in this document follows that of the GUI of the system software. This document does not cover the development environment setup and the development procedure, so refer to the following documents as needed.

Note
Wherever a distinction is not required in this document, the term "wireless controller" will be used in reference to the SIXAXIS™ wireless controller, the DUALSHOCK®3 wireless controller and the DUALSHOCK®4 wireless controller.

Related Documents

- DevKit/TestKit Setup Guide
- Development Kit Neighborhood Settings Guide

Main Screens

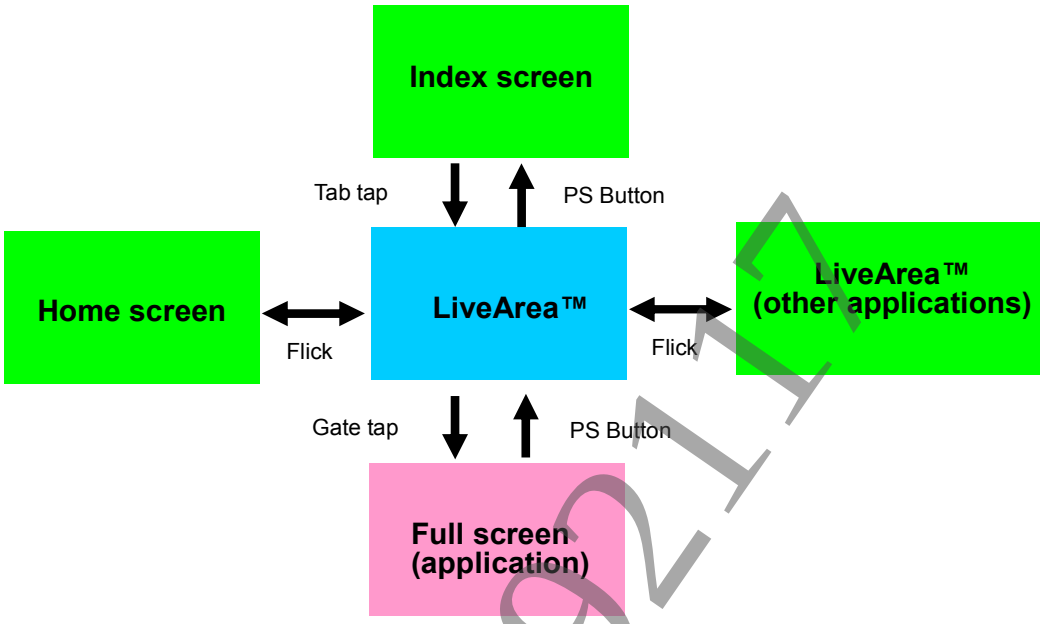
Typical screens of the system software are the following.

- Home screen
  This screen displays the application icons. Tapping an application icon displays LiveArea™ of that application.
- LiveArea™
  This screen displays the application-related information. Applications are basically started up via LiveArea™. Multiple LiveArea™ may exist and can be navigated with a horizontal flick. Home screen is also accessed with a flick.
- Full screen
  This screen is where the application runs.
- Index screen
  This is the mode for displaying home screen and multiple LiveArea™ as a list. Tapping a specific LiveArea™ or "HOME" causes the screen to change to that screen.

Screen Transitions

The screen transitions of the system software are as shown in Figure 1.

Figure 1 Screen Transitions of PlayStation®Vita



If the **PS** button is pressed while an application is running in full screen mode, the screen will switch to LiveArea™. At this time, the application's process will be suspended.

When gate is tapped, switching from LiveArea™ to full screen mode, the application's process will be resumed.

When the operation to peel the application's LiveArea™ is performed (by dragging the upper-right corner), the application's process will be killed.

Home Screen

Home screen is the first screen that is displayed following start-up and is where applications are started up.

The home screen of the DevKit/TestKit is a simplified version of the home screen of the retail unit.

When the DevKit is operating in **Development Mode**, **DevMode** is displayed at the upper left of the home screen. For details on DevKit boot parameters, refer to the "Boot Parameters - Release Check Mode" section in Chapter 4 "★Debug Settings Functions".

Figure 2　Home Screen of DevKit (Development Mode)

The following main icons are described.

- Browser
  This icon starts up the Internet Browser application. For details, refer to the "Internet Browser" section.
- Trophies
  This icon starts up the Trophy application. For details, refer to the "NP Trophy Library Overview" document.
- Friends
  This icon starts up the Friends application. For details, refer to the "Friends Application" section.
- Messages
  This icon starts up the Messages application. For details, refer to the "NP Message Overview" document.
- Party
  This icon starts up the Party application. For details, refer to the "NP Party Library Overview" document.
- PS4 Link
  This icon starts up the PS4 Link application. For details, refer to the "PS4 Link Application" section.
- Parental Controls
  This icon starts up the Parental Controls application.
- Music
  This icon starts up the Music application. For details, refer to the "Music Application" section.
- near
  This icon starts up the "near" application. For details, refer to the "near System Overview" document.

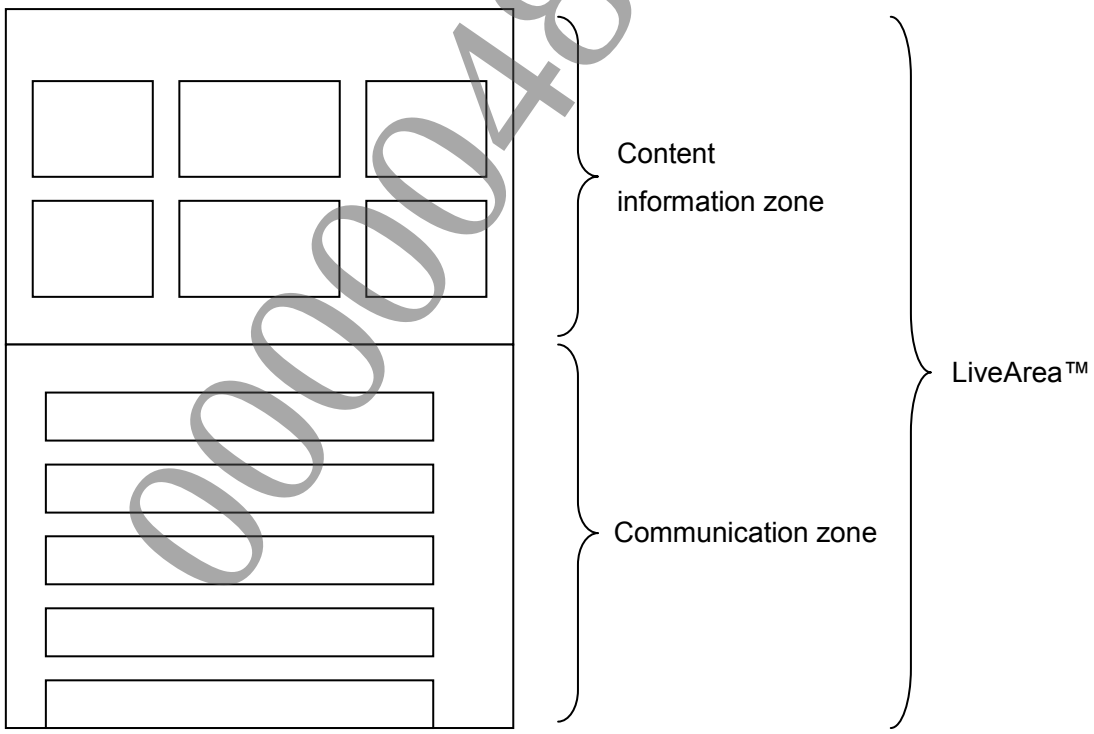
- Photos
  This icon starts up the Photos application. For details, refer to the "Photos Application" section.
- PS3 Remote Play
  This icon starts up the PS3 Remote Play application. For details, refer to the "PS3 Remote Play Application" section.
- Content Manager
  This icon starts up the Content Manager application. For details, refer to Chapter 5 "Functions of Application".
- Settings
  This icon starts up the Settings application. For details, refer to Chapter 3 "Setting Functions" and Chapter 4 "★Debug Settings Functions".
- ★APP_HOME
  This icon starts up the application placed on the development host computer. For details, refer to Chapter 2 "Starting Application from ★APP_HOME".

LiveArea™

LiveArea™ consists of two screens that are changeable by flicking vertically.

- Content information zone
  This area displays the application-related information. For details, refer to the "Content Information Zone" chapter in the "LiveArea™ Specifications" document.
- Communication zone
  This area is for application-related communication. For details, refer to the "Activity System Overview" document.

Figure 3 LiveArea™ Screen Configuration



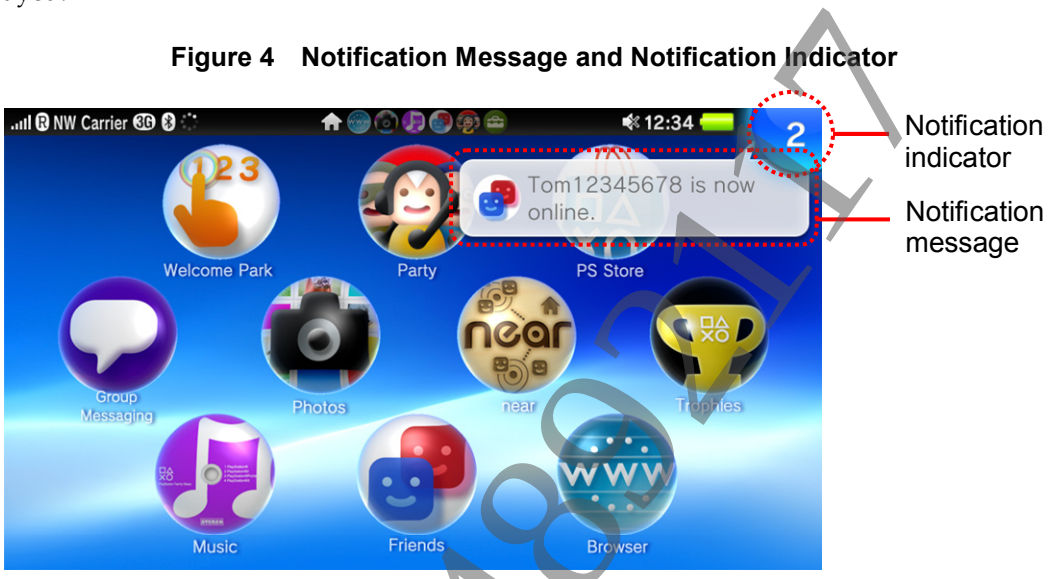
Message Display by the System Software

When necessary, the system software will display on-screen messages. These are called notification messages and system messages, and notify application events, display overall system status, etc.

Notification Message

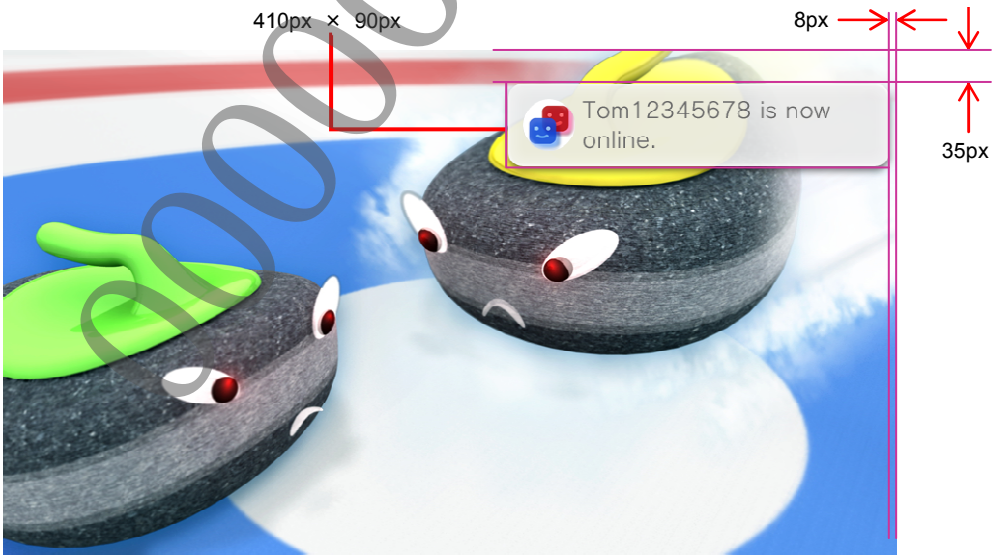
Notification messages notify application events. When a new event occurs, it will be displayed for 3 seconds on the upper right of the screen. If there are multiple new events, each event will be displayed for 2 seconds, and only the last notification message will be displayed for 3 seconds. Also, when there is a notification, the notification indicator will turn blue, and the number of unread notifications will be displayed.

Figure 4 Notification Message and Notification Indicator



The position and size in which notification messages are displayed are as shown in Figure 5:

Figure 5 Notification Message Display Position and Size



Note
Notification messages will be displayed on the upper right of the screen while the application is being displayed in full-screen mode as well. For this reason, bear in mind their position when placing information or buttons required for the application's execution.

Applications such as Messages, Friends, Party, Trophy and "near" notify notification message events.

In terms of download-related notifications, download in progress, download failure, paused download, and download and installation of application system update data and applications are also notified.

System Message

System message is a feature for displaying system status irrespective of specific applications.

System messages disappear after being displayed for 5 seconds. However, if multiple events occur simultaneously, each event will be displayed for 2 seconds, and only the last system message will be displayed for 5 seconds.

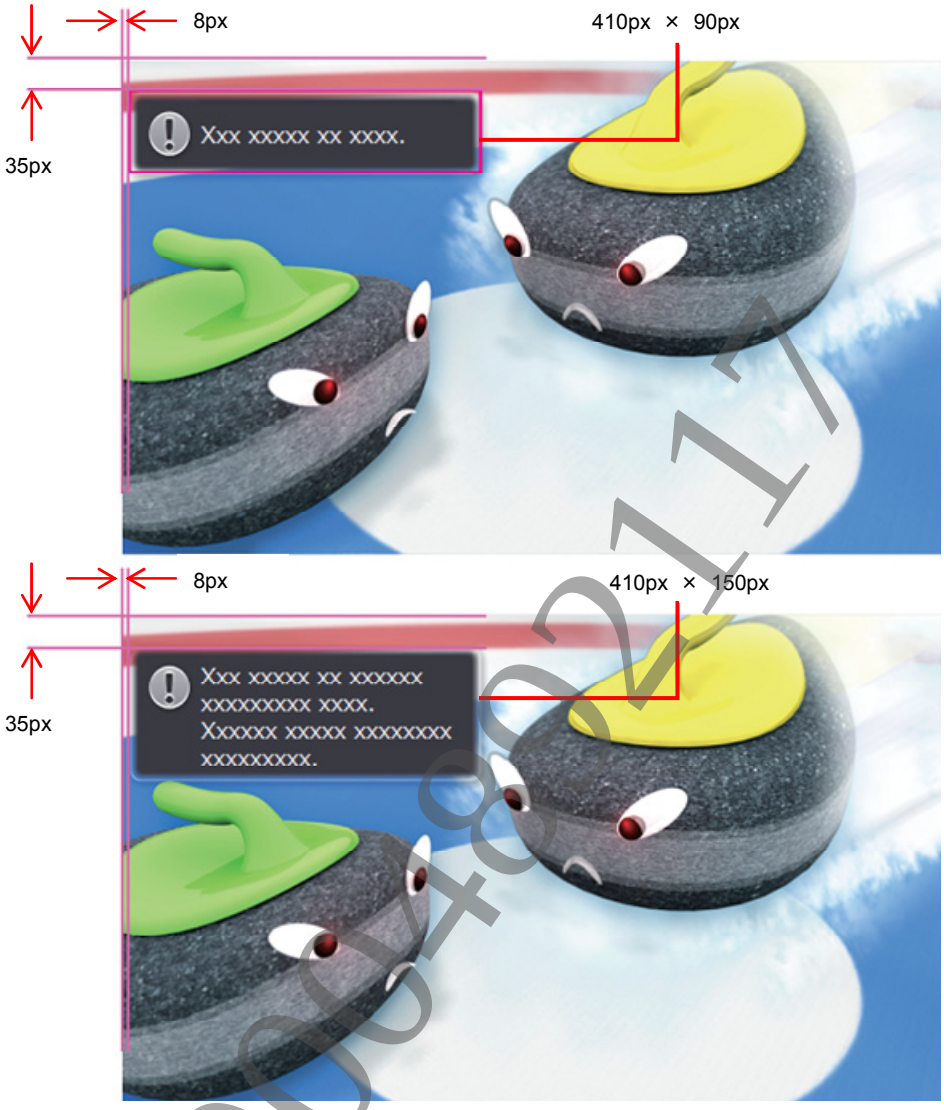
System messages include, for example, memory card-related information, such as insufficient free space, and messages related to battery status and invalid operations.

System messages will also be displayed when the application is running in full-screen mode, and applications cannot use them freely.

Figure 6 System Message Display Example

System messages are displayed on the upper left of the screen, as shown in Figure 7.

Figure 7 System Message Display Position and Size



Note
System messages are displayed on the upper left of the screen while the application is being displayed in full-screen mode as well. They are displayed with a vertical size of 90 pixels to a maximum of 150 pixels. For this reason, bear in mind their position when placing information or buttons required for the application's execution.

2 Starting Application from ★APP_HOME

The application saved on the HDD of the development host computer or on the memory card of the DevKit can be started up by tapping the ★APP_HOME icon.

Operation up to Application Start-up

The application cannot be started up right away even by tapping the ★APP_HOME icon. The following operation is required.

(1) Tap the ★APP_HOME icon.
LiveArea™ is displayed.

(2) Tap gate in LiveArea™.
The application on the development host computer starts up.

For details on gate, refer to the "Content Information Zone" chapter in the "LiveArea™ Specifications" document.

Application Save Destination

The save destination of the program file of the application that can be started from ★APP_HOME is one of the following.

- For the HDD of the development host computer
Create the app directory directly under host0: defined in fsroot and rename and save the executable self file as eboot.bin. The file path is host0:app/eboot.bin. Execute the `psp2ctrl update` command or perform **Connect** processing on Neighborhood for PlayStation®Vita to enable referencing the files placed from the DevKit.
- For the memory card
Create the app directory on the memory card, and rename and save the executable self file as eboot.bin. The file path is ux0:data/app/eboot.bin.

When the ★APP_HOME icon is tapped, the system will search for available directories in the following order, and start up the application with the directory that is first found as the target.

(1) host0:app

(2) ux0:data/app

For example, if you wish to start up an application on the memory card, do not prepare the host0:app directory.

Since the name displayed on the ★APP_HOME icon will vary based on the found directory, it is possible to check which directory will be the target of start-up from the name displayed on the icon.

- For host0:app:
★APP_HOME
(host)
- For ux0:data/app:
★APP_HOME
(ux)

In order to create save data, a savedata directory is additionally required. For the directory configurations, refer to the "Application Development Process Overview" document.

System File

In order to start up an application from ★APP_HOME, the application's system file (param.sfo) is essential. Place the file as follows according to the target that ★APP_HOME references.

- host0:app/sce_sys/param.sfo
- ux0:data/app/sce_sys/param.sfo

If app0:'s location is explicitly specified by the file path setting file, the specified location will also need the system file (app0:/sce_sys/param.sfo). At this time, only the product code (TITLE_ID) will be read from the system file referenced by ★APP_HOME, while other parameters will be read from app0:/sce_sys/param.sfo.

For the method to create the system file, refer to the "Param File Editor User's Guide" document that is included in Publishing Tools. If the format of the file is invalid, a system file format error (error code: C0-11479-2/0x80800015) will occur when the application starts up.

LiveArea™ of ★APP_HOME

LiveArea™ must be displayed in order to start up an application. Therefore, the file that defines LiveArea™ must be included among the files that make up the application. In the case of ★APP_HOME, the directory to which the XML defined in LiveArea™ is to be saved is as follows.

host0:app/sce_sys/livearea/
ux0:data/app/sce_sys/livearea/

For details on LiveArea™ customization, refer to the "LiveArea™ Specifications" document.

However, the restriction that LiveArea™ creation from the start of application development is essential makes the development process complex. The system provides the default definition held by the system to display LiveArea™ for application start-up from ★APP_HOME if the file that defines LiveArea™ does not exist. This default definition is the minimum definition required for application start-up. This default definition is used if the following file does not exist.

host0:app/sce_sys/livearea/contents/template.xml
ux0:data/app/sce_sys/livearea/contents/template.xml

Note that even if template.xml does exist, if its contents are invalid, application start-up from LiveArea™ may not possible

Placement of Data on the Memory Card

Below is an explanation on how to place data in ux0:data/app on the memory card.

(1) Create an app directory in the ux0:data directory.

Execute the following command:

```
> psp2ctrl mkdir ux0:data/app
```

(2) Place the necessary data on ux0:data/app.

Place the application, the data read from the application, LiveArea™ definition files, images etc.

For example, if placing the self file xxxx.self, which is the application, as ux0:data/app/eboot.bin, execute the following command:

```
> psp2ctrl cp xxxx.self ux0:data/app/eboot.bin
```

Likewise, place other files using the psp2ctrl cp command and the psp2ctrl mkdir command as appropriate.

3 Setting Functions

Various setting functions related to the operation of the application and system software are provided in **Settings**.

- Flight Mode
- System Update
- Network
- PSN℠
- Devices
- Notifications
- Sound & Display
- Theme & Background
- Location Data
- Security
- Date & Time
- Language
- Accessibility
- System
- Format
- Power Save Settings
- ★Debug Settings

Some of the usable settings vary depending on the device (DevKit or TestKit) used in development and the DevKit boot parameter settings. For information on the usable settings for these devices and for each boot parameter, refer to the "Settings Usable With Each Device for Development and Boot Parameter" section at the end of the chapter.

As **★Debug Settings** in the list above includes system-related setting items and functions for development, it is described separately in Chapter 4 "★Debug Settings Functions".

Precautions

Although the settings of the DevKit (target) can be executed from the development host computer (host), avoid executing simultaneously the setting functions from the DevKit side and the setting functions from the development host computer side (Neighborhood for PlayStation®Vita). (However, starting up setting functions of Neighborhood for PlayStation®Vita is not a problem.) Simultaneously using the setting functions of the Development Kit and the development host computer may impair the system operation owing to discrepant setting values. For details about settings from the host, refer to the "Development Kit Neighborhood Settings Guide" document.

If the system itself or the created programs do not operate in accordance with settings, and settings information is likely to have been damaged, recover settings information. Such damage to settings information may occur if the power supply to the DevKit is interrupted while settings information is being written. To recover settings information, execute the procedure described under "Restore This System" in the "Safe Mode Features and Influence of Their Execution" section of Chapter 6, "Safe Mode Features".

Also, if the Settings application detects that settings information has become corrupted, a system error dialog stating "A serious error has occurred in the system software. This system will restart." will be displayed, and the DevKit/TestKit will be automatically started up in safe mode. Execute the repair processing available for selection from the safe mode menu. Repair is not guaranteed, but system operation may improve. If "Update System Software" is highlighted, we recommend to first select "Restore This System".

Flight Mode

In the Flight Mode, the Wi-Fi and Bluetooth® functions of PlayStation®Vita are disabled collectively. By only setting the **Flight Mode** to **On**, all these communications (radio wave output) are disconnected. Each communication function can be enabled by separately setting the Wi-Fi and Bluetooth® functions to **On** via corresponding setting items after setting the **Flight Mode** to **On**.

When changing each setting of the Wi-Fi and Bluetooth® functions after setting the **Flight Mode** to **On**, the **On/Off** state of these functions will remain unchanged even after resetting the **Flight Mode** to **Off**. Conversely, when not changing each setting of the Wi-Fi and Bluetooth® functions after setting the **Flight Mode** to **On**, the **On/Off** state of these functions is restored to the state before the **Flight Mode** is set to **On**.

System Update

System Update is a function for performing system updates by acquiring the latest update data from a memory card, the development host computer or HTTP server on the network. A system is also provided to save multiple update data, allowing updating by selecting the appropriate update data as needed. For details, refer to the "DevKit/TestKit Setup Guide" document.

Network

This mainly performs settings related to internet connections. For details on the network devices supported by PlayStation®Vita/PlayStation®TV and the connection behavior, refer to the "Network Overview" document.

Internet Connection Status

Connection target setting name, IP address, NAT type etc. are displayed as Internet connection status of PlayStation®Vita/PlayStation®TV at that point in time.

If PlayStation®Vita remains in a state of inactive communication for a given period of time, the system will perform disconnection. Therefore, note that it is possible that status may become disconnected. PlayStation®TV maintains an always-connected state.

Internet Connection Test

Checks connection to the network and communication to the Internet. The most appropriate settings at the time of connection will be used.

It is also possible to individually select settings that have been created from **Wi-Fi Settings/Internet Connection Settings,** and to perform an **Internet Connection Test**. In this case, the test will explicitly use these settings.

Wi-Fi Settings/Internet Connection Settings

To use the Internet communication mode of the network function, network setting must be created beforehand. **Wi-Fi Settings/Internet Connection Settings** is used mainly to make or delete these network settings. A number of additional settings, including On/Off control of the Wi-Fi function, On/Off control of the Internet connection function, and channel setting for the ad hoc communication mode are also supported.

In the **Wi-Fi Settings**, the wireless settings for using the Wi-Fi function and the wired settings for using USB Ethernet (the DevKit/TestKit exclusive) can both be created.

A **LAN Cable** setting for Ethernet is available for the **Internet Connection Settings** connection method on PlayStation®TV. In addition, it is also possible to create a wireless setting that uses the Wi-Fi function.

To create the wired settings with a DevKit/TestKit, select **Add Manually** and select ★**USB Ethernet**. This creates the wired settings.

To create the wireless settings, the scan results are periodically displayed on the top screen of the **Wi-Fi Settings/Internet Connection Settings**, so select the desired access point on that screen and create the settings. In the case of an access point that is not displayed in the scan results, select **Add Manually** and select **Enter Manually**. The screen for inputting the SSID and security method is displayed, so input the appropriate items and then select **OK** to create the wireless settings.

For both the wired and wireless settings, the default values are used for the various created setting items. If manual specification of the IP address is desired, this can be done in **Advanced Settings**, which is described below.

The created network settings are listed on the top screen of **Wi-Fi Settings/Internet Connection Settings**. Selecting a setting displays a corresponding menu, and selecting **Delete** in that menu deletes the corresponding network settings. The various settings can be edited by selecting **Advanced Settings**. Manually changing the IP address is also done in the same location.

The following items can be set in **Advanced Settings**.

- **IP Address Settings**
- **DNS Settings**
- **MTU Settings**
- **Proxy Server**

The creation, deletion, and editing of these network settings is also possible from the development host computer. For details, refer to the "Development Kit Neighborhood Settings Guide" document.

Relationship between Wi-Fi Settings/Internet Connection Settings and Applications

The network connection is automatically performed by the system software as needed. Applications do not perform network connection.

Applications can acquire whether the network is connected to using the API of libnetctl. Connection information including whether the device used for network connection is wired or wireless, as well as the IP address, netmask, BSSID and the security settings, can also be acquired.

The setting contents of **Wi-Fi Settings/Internet Connection Settings** cannot be changed by applications.

PSN℠

PSN℠ has the following functions.

- Sign-up to Sony Entertainment Network
- Sign-in to PSN℠
- Sign-out from PSN℠
- Check and Change Profile
- Check and Change Account Information
- Automatic Update Settings
- Sub Account Management
- System Activation/Deactivation
- Privacy Settings
- Coordination with Facebook®

- Coordination with Twitter

Sign Up to Sony Entertainment Network

The **Sign Up** and **★Quick Sign Up** item for signing up to Sony Entertainment Network is displayed if this has not been already done. When the **Sign Up** item is selected, the application for sign-up starts up, so follow the directions from the application to complete the signup procedure. Upon completion of signup, the main unit enters the signup completion state.

Selecting the **★Quick Sign Up** item allows the creation and signup of a new account by inputting a small number of items. This function is provided exclusively for the DevKit/TestKit.

Sign-in to PSN℠

When sign-up to Sony Entertainment Network has been done and the state is the signed out state from PSN℠, the **Sign In** item for signing in to PSN℠ is displayed. When this item is selected, the main unit enters the signed in state.

Sign-out from PSN℠

When sign-up to Sony Entertainment Network has been done and the state is the signed in state to PSN℠, the **Sign Out** item for signing out from PSN℠ is displayed. When this item is selected, the main unit enters the signed out state.

Relationship between Sign-In and Applications

In order for an application to use the functions (instant message function, matching, ranking service, etc.) provided by PSN℠, sign-in to PSN℠ must have been completed. An application can check whether sign-in to PSN℠ is completed using the API of Network Check Dialog. If sign-in has not been completed, a dialog box prompting the user to sign in is displayed in Network Check Dialog. Automatic sign-in by calling an application is not possible. Neither can applications perform sign-out from PSN℠.

Sign-out may occur while an application is running. If sign-out occurs, the application must always return to the offline phase.

Check and Change Profile

When sign-up to Sony Entertainment Network has been done and the state is the signed-in state to PSN℠, the **Profile** item for checking and changing the profile of PSN℠ can be used. When this item is selected, the profile is displayed and the contents of the various profile items (Avatar, Panel, etc.) can be changed.

Check and Change Account Information

When sign-up to Sony Entertainment Network has been done and the state is the signed-in state to PSN℠, the **Account Information** item for checking and changing the information of the Sony Entertainment Network account is displayed. When this item is selected, the account information is displayed and the various account information contents can be changed.

Automatic Update Settings

The enabled/disabled setting of the function for automatically downloading update files or the function for automatically synchronizing the trophy information with the PSN℠ server when an application update has occurred can be changed.

Only PlayStation®Plus members can enable the automatic update function.

Also refer to the PSN℠ - Fake Plus section.

Sub Account Management

When sign-up to Sony Entertainment Network has been done, the state is the signed-in state to PSN℠, and the account manages subaccounts, the **Sub Account Management** item for setting sub accounts is displayed. When this item is selected, the list of the sub accounts managed by the account is displayed. When a sub account whose settings are to be changed is selected, a screen for changing the management status is displayed.

System Activation/Deactivation

When sign-up to Sony Entertainment Network has been done and the state is the signed-in state to PSN℠, the **System Activation** item for System activation or deactivation of the main unit is displayed. When this item is selected, the screen for selecting the activation type is displayed. When the type is selected, the **Activate** and **Deactivate** buttons are displayed. To activate, select **Activate**, and to deactivate, select **Deactivate**.

Privacy Settings

When sign-up to Sony Entertainment Network has been done and the state is the signed-in state to PSN℠, the **Privacy Settings** item for setting the range of information display to PSN℠ is displayed. This item allows setting of the release range of the friends list and activities.

Coordination with Facebook

The coordination function between PSN℠ and Facebook can be set via the **Facebook** item. With this function, it is possible to bind the Facebook accounts and set **On**/**Off** of each function related to coordination with Facebook. For details on the coordination function with Facebook, refer to the "Facebook® Coordination System Overview" document.

Coordination with Twitter

Settings related to the use of Twitter can be made from **Twitter** item. The following functions are provided.

- Register the Twitter account information to DevKit/TestKit
  If the Twitter account information has not been registered to DevKit/TestKit, tapping **Twitter** item displays the screen for entering the Twitter account information.
  Once the account information has been entered, tap **Sign in** to save the Twitter account information to the DevKit/TestKit.
- Specify tweet function use for individual applications
  Use of the tweet function can be specified separately for each application. This setting is enabled by tapping the check box of the application for which tweet function use is desired to set it to **On**, thereby enabling use of the tweet function for that application.
- Delete registered account information
  The Twitter account information registered to DevKit/TestKit can be deleted.

Devices

This performs settings related to peripherals

Wireless Controllers

On PlayStation®TV, it is possible to change the controller number of a registered wireless controller and to enable/disable the vibration feature.

Bluetooth® Settings

In order to use a Bluetooth® device with PlayStation®Vita/PlayStation®TV, that device must be registered beforehand in the **Bluetooth® Settings**. In the **Bluetooth® Settings**, registration, deletion, connection and disconnection of a Bluetooth® device such as headsets, microphone, headphones, speaker, and keyboards can be done. Moreover, **On/Off** switching of the Bluetooth® feature can be carried out on PlayStation®Vita, and a setting can be made to prioritize Bluetooth® device microphones over the microphone of PlayStation®Vita.

Bluetooth® devices can be registered by searching the Bluetooth® devices that are in the active state, selecting the device to be registered from the displayed list, and inputting its pass key or confirming its registration.

The registered Bluetooth® devices are listed on the top screen of the **Bluetooth® Settings**. Selecting a Bluetooth® device changes the screen to the operation screen, where deletion, connection, and disconnection can be done according to the communication state of the Bluetooth® device.

Register BD Remote Control

On PlayStation®TV, a BD remote control can be registered by following the instructions onscreen.

External Keyboard

The type of external keyboard and key repeat (time to start/speed) can be set.

When selecting Japanese or Chinese - Traditional for type, the input method can be set.

Notifications

This setting allows the notification alerts function to be enabled/disabled. This setting can be made individually for each application that sends out notifications.

Enable Notification Alerts

Select this to enable the notification alerts function.

When **Enable Notification Alerts** is deselected, the notification alerts function is disabled for all applications.

When this setting is enabled, the following settings can be enabled for each application that sends out notifications.

Show on Screen

If this is selected, a notification message will be displayed on the screen when a notification arrives.

Light Up PS Button/Light Up Notification Indicator

If this is selected, the PS button (notification indicator) will flash when a notification arrives.

Play Sound

If this is selected, a sound will be played when a notification arrives.

Relationship between Notifications and Applications

An application cannot retrieve or change Notifications settings.

Sound & Display

Sound & Display includes the following settings related to sound and display.

Resolution

Changes the video output resolution for the HDMI port.

RGB Range

Changes the video RGB range setting.

Display Area Settings

Allows adjusting of the size of the image shown on the display.

Start Screensaver

Allows changing of the time until the screen saver starts functioning in a non-operating state and allows disabling of the screen saver function.

Brightness

Adjusts display brightness.

System Music

If this is selected, a BGM will play in LiveArea™.

AVLS

Select this to enable the function for headphone volume control.

Note
On some types of retail units, when the accumulated playback time exceeds a certain time while this function is disabled, the function will automatically become enabled again.

Mute Automatically

If this is selected, the audio from the body's speakers is muted automatically when the headphones are separated from the headset jack.

Relationship between Sound & Display and Applications

An application cannot retrieve or change Sound & Display settings.

Theme & Background

Features are provided to apply system software themes and to change the start screen and home screens.

Theme

Select one system software theme from those installed, and apply it to the home screens and start screen. For details on system software themes, refer to the "System Software Theme Overview" document.

Start Screen

Changes the design of the start screen.

Home Screen Backgrounds

Changes the backgrounds of home screens.

Location Data

Location Data includes **Use Location Data** and settings for each application. With these settings, it is possible to enable/disable application to use current location data retrieved from GPS, Wi-Fi or 3G base stations.

When **Use Location Data** is deselected, the location information service will be disabled for all applications. When **Use Location Data** is selected, it is possible to allow use of the location information service by application.

Items for each application are added as settings when the location information service is first accessed by that application.

When the item of an application is deselected, use of location data for that application becomes disabled.

Relationship between Location Data and Applications

An application cannot change location data settings directly. However, the user can change location data settings through the dialog displayed with `sceLocationConfirm()`, an API of liblocation.

Security

Screen Lock

Screen Lock includes the 2 items: **Confirm passcode when unlocking** and **Change Passcode**.

If **Confirm passcode when unlocking** is selected, a screen for inputting the screen lock passcode will be displayed when cold booting or returning from sleep state.

In **Change Passcode**, set the passcode necessary when modifying the **Confirm passcode when unlocking** setting or releasing screen lock.

Relationship between Security and Applications

An application cannot retrieve or change security settings.

Date & Time

Date & Time includes the **Date & Time Settings**, **Time Zone**, **Adjust Daylight Saving Automatically**, **Date Format**, and **Time Format** settings.

Setting of the current time is done in **Date & Time Settings** and **Time Zone**. The time and date according to the daylight saving time can also be made to be displayed with the **Adjust Daylight Saving Automatically** setting. The display method for the current time can be set in **Date Format** and **Time Format**.

Item	Choices
Date Format	YYYY/MM/DD, DD/MM/YYYY, MM/DD/YYYY
Time Format	12-hour, 24-hour
Time Zone	Time zone of major cities

Relationship between Date & Time and Applications

An application can acquire the contents of date & time settings by using the API of application utility library, as well as the current time and date by using the API of librtc. In addition to the current date and time of the set time zone, an application can acquire the current time and date of any time zone. An application cannot change the current time and date.

Language

System Language

System language can be selected from 20 languages: Dansk, Deutsch, English (United Kingdom), English (United States), Español, Français, Italiano, Nederlands, Norsk, Polski, Português (Brasil), Português (Portugal), Русский, Suomi, Svenska, Türkçe, 日本語, 한국어, 简体中文, and 繁體中文. Text such as those displayed by the system software will be switched to the selected language.

Input Languages

It is possible to enable/disable on-screen keyboards by language, to set input functions, and to register/edit frequently used vocabulary.

> **Note**
> The on-screen keyboards of all the languages cannot be disabled at the same time.

Relationship between Language and Applications

System Language settings can be retrieved by using the API of the application utility library.

Accessibility

This performs settings related to accessibility.

Zoom

When this item is checked, the zoom feature can be used via controller operations.

Invert Colors

When this item is checked, colors of the screen display will be inverted.

Larger Text

When this item is checked, texts will be displayed larger.

Bold Text

When this item is checked, texts will be displayed in bold fonts.

High Contrast

When this item is checked, colors of the screen display will be enhanced.

Closed Captions

This setting is the same as the closed captions setting in the video player. When the closed caption is changed here, the closed caption setting of the video player will also change.

Closed captions can be enabled or disabled, and other detailed settings can be made.

Button Assignments

Certain button assignments can be changed to arbitrary buttons. Assignment setting can be enabled or disabled, and assignment settings can be made.

L/R sticks can only be switched, and only Controller Number 1 can be a target for PlayStation®TV.

Add to Quick Menu

When this item is checked, **Invert Colors** and **Enable Custom Button Assignments** setting items for Accessibility will be added to the quick menu.

System

System Information

This function displays the version of the system software, the MAC addresses of the main unit's Wi-Fi device and USB Ethernet adapter and the memory card capacity and its free space.

Auto-Start Settings

This menu item sets the application(s) or function(s) that are to start automatically. When this setting is set to **Off**, the application or function does not start automatically.

Control with Buttons on This System

When this setting is set to **On**, operation with the buttons on the PlayStation®Vita unit will be possible. There are some operations that are not possible with buttons even when this setting is set to **On**.

PlayStation®Mobile

PlayStation®Mobile runtime packages can be updated and deleted. Tap the **options** button and select **Update** or **Delete** from the menu that appears.

Enable Turning On This System from Network

This allows the DevKit power to be turned on from other devices through a network.

Enable HDMI Device Link

This allows linked processing such as the power control between HDMI devices.

Media Forwarding Settings

This performs settings when video is played back from another device using media forwarding protocols through a network.

Error History

This is a function to review serious errors occurred during the execution of the system software. The 10 most recent errors are displayed.

Intellectual Property Notices

This displays the intellectual property notices of the programs, etc., that are included in the system software.

Relationship between About System and Applications

An application cannot acquire the version of the system software. The version of the system software required by the application is embedded in param.sfo, and version check is executed at start-up. In other words, when the application has started up, the operation of the system software of that version or later is guaranteed.

Use the API of libnet to retrieve the MAC address (Ethernet address) of the main unit's Wi-Fi device.

The application cannot record errors directly onto the **Error History**. If a serious error occurs during the execution of system software functions called by the application, such as Common Dialog, it will be recorded indirectly.

Format

Clear Learning Dictionary

Deletes learning dictionary, resetting the display order of vocabulary in word prediction dictionaries, etc.

Format Memory Card

This menu item formats the memory card.

Restore Settings

The system software-related settings are restored.

Restore This System

This function restores PlayStation®Vita/PlayStation®TV, restoring all the setting values in the internal flash memory. The data in the internal flash memory is deleted and the state returns to the initial state at the time of purchase.

Relationship between Format and Applications

An application cannot restore these items.

Power Save Settings

Enter Standby Mode Automatically

It is possible to set the time taken for PlayStation®Vita to automatically switch from non-operating state to standby state.

When the set time elapses without any user operation or any particular specification from the application, PlayStation®Vita will switch to standby state.

Note
★ **Off** and ★ **After 60 Minutes** are the options provided only for the DevKit/TestKit for debugging purposes. Note that the operation of **Enter Standby Mode Automatically** cannot be set to **Off** in the retail unit.

Note
Irrespective of this setting, dimmed display operation is executed after a fixed time elapses.

Turn Off Wireless Controllers Automatically

This allows the setting of the time until the power of wireless controllers is turned off in a non-operating state. When set to **Off**, the function for automatically turning off the power will be disabled.

Use Wi-Fi in Power Save Mode

When this is checked, Wi-Fi can be used in the power save mode.

Settings Usable With Each Device for Development and Boot Parameter

The usable settings vary depending on the device (DevKits and TestKits) used in development and DevKit boot parameter settings. A summary of the usable settings for these devices and each of the boot parameters is as follows.

Setting Item	DevKit		TestKit	
	PS TV Emulation		PTEL-1000 Series	PTEL-2000 Series
	Off	On		
System Update	Yes	Yes	Yes	Yes
Update Using Wi-Fi	Yes		Yes	Yes
Update by Connecting to a PS3™ System	Yes		Yes	Yes
Update by Connecting to a PC	Yes		Yes	Yes
Network	Yes	Yes	Yes	Yes
Wi-Fi Settings	Yes		Yes	Yes
Internet Connection Settings		Yes		
Internet Connection Test	Yes	Yes	Yes	Yes
Internet Connection Status	Yes	Yes	Yes	Yes
Devices	Yes	Yes	Yes	Yes
Wireless Controllers		Yes		
Bluetooth® Devices	Yes	Yes	Yes	Yes
Register BD Remote Control		Yes		
External Keyboard	Yes	Yes	Yes	Yes
PSN℠	Yes	Yes	Yes	Yes
Sign Up/Sign In/Sign Out	Yes	Yes	Yes	Yes
Profile	Yes	Yes	Yes	Yes
Account Information	Yes	Yes	Yes	Yes
Automatic Update Settings	Yes	Yes	Yes	Yes
Sub Account Management	Yes	Yes	Yes	Yes
System Activation	Yes	Yes	Yes	Yes
Privacy Settings	Yes	Yes	Yes	Yes
Facebook	Yes	Yes	Yes	Yes
Twitter	Yes	Yes	Yes	Yes
Notifications	Yes	Yes	Yes	Yes
Enable Notification Alerts	Yes	Yes	Yes	Yes
Show on Screen	Yes	Yes	Yes	Yes
Light Up PS Button	Yes		Yes	
Light Up Notification Indicator				Yes
Play Sound	Yes	Yes	Yes	Yes
Sound & Display	Yes	Yes	Yes	Yes
Resolution		Yes		
RGB Range		Yes		
Display Area Settings		Yes		
Start Screensaver		Yes		
Brightness	Yes		Yes	Yes
System Music	Yes	Yes	Yes	Yes
AVLS	Yes		Yes	Yes
Mute Automatically	Yes		Yes	Yes
Theme & Background				
Theme	Yes	Yes	Yes	Yes
Start Screen	Yes		Yes	Yes
Home Screen Backgrounds	Yes	Yes	Yes	Yes

Setting Item	DevKit		TestKit	
	PS TV Emulation		PTEL-1000 Series	PTEL-2000 Series
	Off	On		
Location Data	Yes		Yes	Yes
Use Location Data	Yes		Yes	Yes
Security	Yes		Yes	Yes
Screen Lock	Yes		Yes	Yes
Date & Time	Yes	Yes	Yes	Yes
Date & Time Settings	Yes	Yes	Yes	Yes
Time Zone	Yes	Yes	Yes	Yes
Adjust Daylight Saving Automatically	Yes	Yes	Yes	Yes
Date Format	Yes	Yes	Yes	Yes
Time Format	Yes	Yes	Yes	Yes
Language	Yes	Yes	Yes	Yes
System Language	Yes	Yes	Yes	Yes
Input Languages	Yes	Yes	Yes	Yes
Accessibility	Yes	Yes	Yes	Yes
Zoom	Yes	Yes	Yes	Yes
Invert Colors	Yes	Yes	Yes	Yes
Larger Text	Yes	Yes	Yes	Yes
Bold Text	Yes	Yes	Yes	Yes
High Contrast	Yes	Yes	Yes	Yes
Closed Captions	Yes	Yes	Yes	Yes
Button Assignments	Yes	Yes	Yes	Yes
Add to Quick Menu	Yes	Yes	Yes	Yes
System	Yes	Yes	Yes	Yes
System Information	Yes	Yes	Yes	Yes
Auto-Start Settings	Yes	Yes	Yes	Yes
Control with Buttons on This System	Yes		Yes	Yes
USB Power Supply	Yes		Yes	Yes
PlayStation®Mobile	Yes	Yes	Yes	Yes
Enable Turning On This System from Network		Yes		
Enable HDMI Device Link		Yes		
Media Transfer Settings		Yes		
Error History	Yes	Yes	Yes	Yes
Intellectual Property Notices	Yes	Yes	Yes	Yes
Format	Yes	Yes	Yes	Yes
Clear Learning Dictionary	Yes	Yes	Yes	Yes
Format Memory Card	Yes	Yes	Yes	Yes
Internal Memory Card				Yes
Removable Memory Card				Yes
Restore Settings	Yes	Yes	Yes	Yes
Restore This System	Yes	Yes	Yes	Yes
Power Save Settings	Yes	Yes	Yes	Yes
Enter Standby Mode Automatically	Yes	Yes	Yes	Yes
Turn Off Wireless Controllers Automatically		Yes		
Use Wi-Fi in Power Save Mode	Yes	Yes	Yes	Yes

4 ★Debug Settings Functions

★Debug Settings, which is one of the **Settings** items, is provided only for the DevKit/TestKit and includes various functions for development support.

Some of the usable settings vary depending on the device (DevKits and TestKits) used in development and DevKit boot parameter settings (**PS TV Emulation** and **Release Check Mode)**. For the usable settings for these devices and each boot parameter, refer to the "Debug Settings Usable With Each Device for Development and Boot Parameter" section at the end of the chapter.

Precautions

Basically only English is supported for the text of the setting items names and the choices in **Debug Settings**.

Boot Parameters - Release Check Mode

This is a setting related to the boot parameters of the DevKit. If this setting is set to **Release Mode**, programs will be executed in an environment equivalent to that of the retail unit. Access to the host file system and debugging functions, which cannot be implemented in the retail unit environment, are not available. Memory size is also the same as in retail units. During normal development, specify **Development Mode**.

When **Development Mode** is specified, **DevMode** will be displayed on the upper left of the home screen.

Note
When this setting is set to **Release Mode**, the items that can be set under the **Debug Settings** are limited. For details, refer to the "Debug Settings Usable With Each Device for Development and Boot Parameter" section.

Boot Parameters - Release Mode Console

This is a setting related to the boot parameters of the DevKit. Even if **Boot Parameters - Release Check Mode** is set to **Release Mode**, in some cases output to the console will be necessary for debugging purposes. If this setting is set to **Enable**, standard output and standard error output will be output to the console of the development host computer. However, conduct the final check of the application's operation with this setting set to **Disable**. This setting item is available for setting only when **Boot Parameters - Release Check Mode** is **Release Mode**.

Note
If a large volume of data is output to a console with Release Mode Console function, system performance may decrease, causing sound skip or other problems.

Boot Parameters - Memory Size

This is a setting related to the boot parameters of the DevKit. If this setting is set to **Console Size**, programs will be executed with the same memory size as in retail units. In addition to the memory size being changed to the retail unit size, access to the host file system and debugging functions will be enabled, as when **Boot Parameters - Release Check Mode** is **Development Mode**. Furthermore, if **Boot Parameters - Release Check Mode** is **Release Mode**, memory size will be limited to that of retail units regardless of this memory size setting.

Boot Parameters - PS TV Emulation

When this setting is **On**, DevKit operation will be the same as PlayStation®TV. When this setting is **Off**, DevKit operation will be the same as PlayStation®Vita.

Note
Every time the setting is changed, settings with region-dependent default values will be initialized.

System Update - Update Server URL

This menu item allows you to switch the system's behavior if you have chosen **Update Using Wi-Fi** or **Update by Connecting to a PC** in the **System Update** settings menu. Update data will be stored on the HTTP server in the local network, enabling system update via the network.

Set the update list file URL on the HTTP server in this item, referring to "Update Using Wi-Fi" in the "DevKit/TestKit Setup Guide" document.

If **Show PUP List** is set to **Off**, only the first of the update data in the update list file will be referenced; if it is **On**, all of the update data in the update list file will be listed up.

There is no default setting. Selecting **Update Using Wi-Fi** in default state will result in the error "One or more settings are not valid.".

System Update - Show PUP List

This menu item allows you to switch the system's behavior if you have chosen **Update Using Storage Media**, **Update Using Wi-Fi** or **Update by Connecting to a PC** in the **System Update** settings menu. By setting this item to **On**, a list of up to 20 update data items is displayed, and the applicable update data can be selected from this list. **Off** is set by default.

If **Update Using Storage Media** is selected, directories will be searched in the following order: memory card (ux0:data/PSP2/UPDATE/SEARCH), and the development host computer (host0:/PSP2/UPDATE/SEARCH). All update data saved immediately under the directory that is found first will be listed on the screen.

When **Update Using Wi-Fi** or **Update by Connecting to a PC** is selected, all the update data in the update list file are displayed in list form on the screen.

System Update - Enable Host0

This menu item allows you to switch the system's behavior if you have chosen **Update Using Storage Media** in **System Update** of the settings menu. When the setting value is **Enable** (default), update data on the development host computer will be referenced in addition to the memory card. When set to **Disable**, update data on the development host computer will not be referenced. If **Show PUP List** is set to **Off**, host0:PSP2/UPDATE/PSP2UPDAT.PUP will be browsed, while if set to **On**, all the update data immediately under the host0:PSP2/UPDATE/SEARCH directory will be listed.

Network - Network Emulation

This menu item can be set for network emulation execution. When **Disabled** is selected in the top list, network emulation is disabled. When one of values **1** to **4** is selected, network emulation is enabled with the parameter corresponding to that value.

For the meaning of each value, refer to the "libnet Overview" document. In the bottom list, select the interface for setting network emulation.

This setting is valid for Internet communication mode, ad hoc communication mode, and PSPNET ad hoc communication mode.

Network - Intermittent Connection

This menu item allows changing the operation of intermittent connection for debugging purposes. When **On** is selected on the list, intermittent connection will be enabled. When **Off** is selected, intermittent connection will be disabled. At the time of title release, intermittent connection operates in enabled state.

For the intermittent connection, refer to the "Network Overview" document.

Network - Fake 3G Interface

This menu item is a function for giving the appearance of using a 3G line from the application on the Development Kit. Set this item to **On** to enable the function.

For details, such as what APIs in particular are faked by the application, refer to the "Network Overview" document.

Network - NAT Traversal Information

This menu item is used to display the information about the NAT Traversal function.

For details on the NAT Traversal function, refer to the "NP Matching 2 System Overview" document.

Network - PSP Adhoc SSID prefix

This menu item sets the first 3 characters of the SSID to be used in PSPNET Ad hoc communication mode.

PSPNET Ad hoc communication will only be possible between devices with matching values for this menu item. The default is “PSP”.

PSN℠ - Account Selection

This menu item is used to switch Sony Entertainment Network accounts. It allows you to add new entries, delete existing entries, and switch to a different entry. Up to 64 entries can be registered, and one account per entry can be bound. Account information such as the sign-in ID and the NP Environment are held for each entry.

Each of the 64 entries is allotted an entry number (user id) from 00 to 63. This entry number indicates the save destination of application data relating to each account, such as the save destination of save data (ux0:/user/<user id>/savedata/<title id>). When a new entry is added, the lowest out of the available entry numbers is adopted as the new entry's entry number; entries are listed in ascending order based on these numbers. Entry numbers can be checked from the last two figures (account ##) of Account Name's initial value. Therefore, we recommend leaving the last 2 figures as they are when changing the Account Name.

Concerning application data relating to the accounts, refer to Chapter 5 "Functions of Application".

PSN℠ - Account Name

This menu item is set as a label for identification when changing the Sony Entertainment Network account in use. It is possible to set an arbitrary character string of 31 characters or less.

PSN℠ - NP Environment

This menu item is a function to set the environmental variables of PSN℠. At present, you should not change this value.

PSN℠ - Fake Plus

This is a debug setting item for checking the PlayStation®Plus operation.

When it is set to **ON**, the user is always treated as a PlayStation®Plus member without checking the right information of the signed-up user account.

When it is set to **OFF**, the operation becomes the same as that of a retail unit, and whether the signed-up user account is a PlayStation®Plus member is checked.

PSN℠ - NP Debug

This menu item is a function for supporting the development of applications that use PSN℠.

The following functions can be switched **On/Off**.

Display of Commerce-Related IDs Specified by the Application

This function displays commerce-related IDs specified by the application when calling an API in the system message. For example, if the service ID specified when calling `sceNpAuthCreateStartRequest()` in the NP Auth library is SP1234-ABCD12345_00, this will be displayed as "Service ID: SP1234-ABCD12345_00". Relevant APIs are as follows:

Library	Function	Display Example
NP Auth	`sceNpAuthCreateStartRequest()`	`NpAuth: SP1234-ABCD12345_00`
NP IN-GAME Commerce 2	`sceNpCommerce2 GetCategoryContentsStart()`	`NpCommerce2 GetCategoryContents: SP1234-ABCD12345_00-CATG000011112222`
	`sceNpCommerce2 GetProductInfoStart()`	`NpCommerce2 GetProductInfo: SP1234-ABCD12345_00-0000111122223333`
	`sceNpCommerce2 GetProductInfoListStart()`	`NpCommerce2 GetProductInfoList: SP1234-ABCD12345_00-0000111122223333`
	`sceNpCommerce2 StartEmptyStoreCheck()`	`NpCommerce2 EmptyStoreCheck: SP1234-ABCD12345_00-CATG000011112222`
Application Utility	`sceAppUtilStoreBrowse()` (When *type* is `SCE_APPUTIL_STORE_BROWSE_TYPE_PRODUCT`)	`TitleStoreApp: SP1234-ABCD12345_00-0000000011112222`
	`sceAppUtilStoreBrowse()` (When *type* is `SCE_APPUTIL_STORE_BROWSE_TYPE_CATEGORY`)	`TitleStoreApp: SP1234-ABCD12345_00-CATG000011112222`
Store Checkout Dialog	`sceStoreCheckoutDialogInit()` (When *mode* is `SCE_STORE_CHECKOUT_DIALOG_MODE_DOWNLOAD2`)	`StoreCheckotDialog Checkout: SP1234-ABCD12345_00` *No system message display when the SKU ID is specified

*When the SKU ID is specified for the Store Checkout Dialog library, the specified list of SKU IDs will be displayed in a dialog format instead of a system message. (When *mode* is `SCE_STORE_CHECKOUT_DIALOG_MODE_CHECKOUT2` or `SCE_STORE_CHECKOUT_DIALOG_MODE_DOWNLOAD2`.)

Display of NP Communication ID Specified by Application

This is a function to display the NP Communication ID specified by the application when an API is called in the system message. For example, if the NP Communication ID specified when the API of the NP Lookup library is called is ABCD12345_00, "NpLookup:ABCD12345" is displayed. The corresponding API is as follows.

Library	Function
NP Lookup	`sceNpLookupCreateTitleCtx()`
NP WordFilter	`sceNpWordFilterCreateTitleCtx()`
NP Message Dialog	`sceNpMessageDialogInit()`
NP Message	`sceNpMessageInit()`
	`sceNpMessageInitWithParam()`
NP Basic	`sceNpBasicRegisterInGameDataMessageHandler()`

Library	Function
NP Trophy	`sceNpTrophyCreateContext()`
NP Matching 2	`sceNpMatching2CreateContext()`
NP Title User Storage	`sceNpTusCreateTitleCtx()`

NP Title Small Storage	`sceNpTssGetSmallStorage()`
	`sceNpTssGetSmallStorageAsync()`
	`sceNpTssGetData()`
	`sceNpTssGetDataAsync()`

NP ScoreRanking	`sceNpScoreCreateTitleCtx()`
NP Party	`sceNpPartyInit()`

Display of Tag ID of Patch

This is a function to display a tag ID given through TPPS (Title Patch Publishing System) that is referenced for checking for the patch in the system message. The Tag ID is displayed as, for example, "TPPS Tag: ABCD12345_T1".

Display of Parameters of the "near" Utility

This function displays the parameters specified by the API when using the "near" utility.

- When the library is initialized

```
NearUtil:NPWRxxxx_00, 0x00000001  // Version of gift data
```

- When gift data is set using the library

```
NearUtil:
   CID=NPWRxxxx_00       // NpCommunicationId
   Ver=0x00000001        // Gift data version
   GID=0x40000002        // Gift ID
   Rad=0                 // Radius within where gift can be found
   Dur=0                 // Valid duration of gift
   Prb=100               // Probability of gift being found
   Rec=0x00              // User attributes that can find the gift
   Unt=0xffffffff        // Number of gift distributions
```

Transaction Forced Delay

This function is for forcibly inserting a delay for the API that accesses the NP server. Use this function to test the operation of an application when the server response is slow due to high load. The supported library is as follows.

- NP Lookup library (a 1 second delay is inserted per request)
- NP WordFilter library (a 1 second delay is inserted per request)
- NP Title User Storage library (a 2-second delay is inserted per request)
- NP ScoreRanking library (a 2-second delay is inserted per request)

Display Indicating the Application Used a PSN℠ Web API That Requires Server Registration

If a PSN℠ Web API requiring server registration is called with the NpWebApi library, the following will be displayed as a system message. Applicable PSN℠ Web APIs are as follows.

Service Name	Display Example
inGamePresence	`NpWebApi inGamePresence`
sessionInvitation	`NpWebApi sessionInvitation`
gameCustomData	`NpWebApi gameCustomData`

If server registration has not been completed, display will be as follows. Depending on the registration status, the service name or the ApiGroup name will be displayed.

Service Name	Display Example
inGamePresence	`NpWebApi inGamePresence Error: ServiceNotConfigured`
sessionInvitation	`NpWebApi sessionInvitation Error: ServiceNotConfiguread`
gameCustomData	`NpWebApi gameCustomData Error: ServiceNotConfigured`

ApiGroup Name	Display Example
userProfile	`NpWebApi (ApiGroup: userProfile) Error: ServiceNotConfigured`
sessionInvitation	`NpWebApi (ApiGroup: sessionInvitation) Error: ServiceNotConfiguread`
gameCustomData	`NpWebApi (ApiGroup: gameCustomData) Error: ServiceNotConfiguread`

PSN℠ - In-Game Commerce Debug

This menu item is a function for testing the operation of an application that uses the in-game browse function of the NP IN-GAME Commerce 2 library. There are three options under this item: **Off**, **Fake SKU and Metadata**, and **Empty Store**. The behaviors of the NP IN-GAME Commerce 2 library functions change as follows when this item is set to **Fake SKU and Metadata** or **Empty Store**.

SKU ID Test

SKU IDs can be retrieved by calling `sceNpCommerce2GetGameSkuInfoFromGameProductInfo()`, but when this item is set to **Fake SKU and Metadata**, the SKU ID of all the SKUs changes to a character string filled with Zs. If proceeding on to purchase processing even though this item has been set to **Fake SKU and Metadata**, improper implementation such as the use of the fixed SKU ID embedded in the application may occur.

SKU Price Test

The price value without the decimal point can be retrieved by calling `sceNpCommerce2GetGameSkuInfoFromGameProductInfo()`, but this value changes to "9999". If the displayed price does not change even when this setting item is set to **Fake SKU and Metadata**, improper implementation such as the display of the fixed value embedded in the application may occur.

Product Detailed Information (Long Form) Size Test

`productLongDescription` can be retrieved by calling `sceNpCommerce2GetGameProductInfo()`, `sceNpCommerce2GetGameProductInfoFromContentInfo()` or `sceNpCommerce2GetGameProductInfoFromGetProductInfoListResult()`, but this value is filled with a 4000-byte character string. The last 7 characters are "----END" so that the end of the character string can be checked and all others are "Z".

Whether or not the application is capable of displaying the entire *`productLongDescription`* can be checked by setting this item to **Fake SKU and Metadata**. When this setting item is set to **Fake SKU and Metadata**, *`productLongDescription`* is converted to a 4000-byte character string locally. Note that this function cannot test the behavior of an application when data of 4000 bytes has actually been retrieved from the network.

Testing the Detection Processing of a Store without Distributed Items

When this item is set to **Empty Store**, APIs will behave so that it will be determined that there are no distributed items in the store. Specifically, when `sceNpCommerce2StartEmptyStoreCheck()` is complete and detection processing of a store without distributed items is finished, `SCE_NP_COMMERCE2_STORE_IS_EMPTY` will return to its argument *`is_empty`*. In addition, when `sceNpCommerce2GetCategoryInfo()`, a function that reads out the category information, is called, 0 will return to *`countOfSubCategory`* and *`countOfProduct`*, which are the members of the argument *`categoryInfo`* (*`SceNpCommerce2CategoryInfo`* structure).

Ensure that the application displays appropriate messages concerning the fact that there are no items for distribution when this item is set to **Empty Store**.

PSN℠ - Service ID

This menu item is used to set the title to be previewed with **PlayStation®Store Preview** of the home screen. Input the issued service ID (example: IV0002-NPXS00004_00).

PSN℠ - Upgradable App Debug

This menu item is a function for supporting the development of upgradable applications. With this setting, the SKU flag values that can be retrieved with `sceAppUtilAppParamGetInt()` will change as shown below. Also, LiveArea™ will change accordingly.

Setting value	SKU flag value
Trial	`SCE_APPUTIL_APPPARAM_SKU_FLAG_TRIAL`
Full	`SCE_APPUTIL_APPPARAM_SKU_FLAG_FULL`
Off	Will not change

If this value is changed during the application operation, the SKU flag value obtained with `sceAppUtilAppParamGetInt()` will immediately change according to the setting value; however, LiveArea™ will only change when it is started up next time.

PSN℠ - Patch Test

This item is for checking behavior when new patches for the game are available. When it is set to **On**, it will return the availability of new patches for the game as a result when checking for patches through Network Check Dialog. Note that this is the behavior occurring when the "sign off flag" in patch server settings is set to "true".

PSN℠ - TPPS Proxy

This is a function to specify the Proxy server to be used to access TPPS (Title Patch Publishing System) for checking for patches. This function supports the Basic authentication.

PSN℠ - Trophy Setup Dialog Debug

This is a debug setting item for checking Trophy Setup Dialog operations.

When it is set to **Always Fail**, it emulates the operation performed when an error occurs during the Trophy Setup Dialog processing. Through this setting, the operation in the case of an error can be checked. If it is set to **Off**, the operation is performed in the same way as in the retail unit.

PSN℠ - Web API Rate Limit Threshold

This is a function to change the limit threshold of PSN℠ Web API call frequencies via the NpWebApi library. A selection can be made from **Always Error**, **10%**, **20%**, … , **90%**, **100%**. Use this function to check the application behavior when the call rate limit is exceeded. For details on the rate limit of PSN℠ Web APIs, refer to the "PSN℠ Web API Overview" document.

PSN℠ - Ignore NpTitleId set by API in Development Mode

When this item is set to **On**, the NP Title ID and NP Title Secret set by the application with the execution of `sceNpSetNpTitleId()` will be ignored even in the **Development Mode**, and the NP Title ID and NP Title Secret stored in the nptitle.dat file will always be used.

Regarding `sceNpSetNpTitleId()`, the NP Title ID, and NP Title Secret, also refer to the "NpWebApi Library Overview" document.

Location Data - Emulate Permission

This menu item is needed for verifying user permission operations for the location information service.

An application cannot use the location information service without the user's permission. This debug setting emulates the user permission status for use of the location information service. When it is set to **On**, the location information service will function based on the permission status set in **Location Data - Permission Status** (below). When it is set to **Off**, the location information service will operate in the same way as in the retail unit.

When the application is started up from the development host computer, or if using the location information service at the time of application start-up from ★APP_HOME, the user permission status for use of the location information service will be emulated regardless of this setting.

Location Data - Permission Status

This menu item allows you to set user permission status for use of the location information service. When an application is started from the development host computer or ★APP_HOME, or when **Location Data - Emulate Permission** is set to **On**, the location information service will function based on this setting. **Disable** indicates that it is not possible to utilize the location information service. **Not Yet Accessed** indicates that the application has not used the location information service yet. **Deny** indicates that use of the location information service is not authorized by the user. **Allow** indicates that use of the location information service is authorized by the user.

Location Data - Device Model

This menu item sets the calculating device for calculating a location.

When set to **System Dependent Model**, the location calculating device implemented on DevKit/TestKit will be used.

When **Emulation Model** is set, the mode will be GPS emulation mode as long as the ux0:/data/gpsdata.nma file exists. The location information service will return calculated information corresponding to the location data written in this file. Moreover, the message by Message Dialog when displaying information that the obtainment of location information failed will be the message normally used for the 3G/Wi-Fi model.

For details on the GPS emulation mode, refer to the "GPS Emulation Function" chapter of the "liblocation Overview" document.

Location Data - Make AP Invisible

This menu item is used to disable Wi-Fi for location calculation while leaving the wireless LAN network enabled, for example during location calculation verification.

When this item is set to **On**, use of the scan information of the AP (Wi-Fi Access Point) for location calculation is forcibly stopped. When this item is set to **Off**, the system's Wi-Fi scan function is used for location calculation.

System - ○ Button Behavior

This menu item allows the assignment of button operations by selecting either circle or cross as **Enter** button. The system software menu will be displayed in accordance with this setting.

Mode	○	×
Enter	Enter	Back
Back	Back	Enter

Button assignment is fixed according to the region.

System - Slow Card Mode

This menu item is used to enable/disable the **Slow Card Mode**.

The **Slow Card Mode** is a function for emulating PlayStation®Vita cards and memory card for which the read/write processing is slow owing to the used device or the internal state. This function is used to conduct tests to create software that will not cause problems in the future.

System - TRC Check Notifications

When this menu item is set, various information for checking whether the application abides by the various TRC regulations and whether it performs operations related to TRC regulations are displayed using system messages. For details, refer to the "Programming Startup Guide" document.

System - GPI Switch

This menu item sets the value of the GPI switch.

It is possible to set an arbitrary eight-digit hexadecimal value. The value set with this item can be retrieved using a kernel API.

For details, refer to the "Kernel Reference" document.

System - HDMI Audio Output

This menu item enables/disables the HDMI audio output.

When this item is set to **Off**, the HDMI audio output is disabled, and the audio is output from the internal speaker or the headphones, etc. instead of the device connected via HDMI. When it is set to **On**, the HDMI audio output is enabled, and the audio is output from the device connected via HDMI.

System - HDMI RGB Full Range

This menu item sets the RGB range of the HDMI video output.

When this item is set to **limited**, output is done in a limited range. When it is set to **Full**, output is done in the full range.

System - PRX Debug

When this item is set to **Enable**, the break points on PRX are enabled by automatically communicating with the debugger on the development host computer when PRX is loaded. The operation of the program, however, is forced to be terminated momentarily, and therefore a sound skip may be caused. When this item is set to **Disable**, some of the break points on PRX will not function, but the occurrence frequency of sound skip can be reduced.

Enable is set by default.

System - Debug Network Clock

This item is used to set the current time to be referenced by `sceRtcGetCurrentNetworkTick()`.

The value set to this item will remain valid until the battery runs out. For details, refer to the "librtc Reference" document.

System - Reset Network Clock

This is a debugging function to confirm the behavior of applications for which the network time has not been initialized due to the battery running out.

This function creates a status that the network time acquired from `sceRtcGetCurrentNetworkTick()` has not been initialized due to the battery running out. After this function is executed, if `sceRtcGetCurrentNetworkTick()` is called when there is not network connection, the function call results in error, returning the error code `SCE_RTC_ERROR_NOT_INITIALIZED`.

In this case, it is recommended that the application launch Network Check Dialog in the PSN℠ mode and prompt the user to sign in. After signing into PSN℠, the network time can be acquired.

System - Deadzone Check of Analog Sticks

The analog sticks of PlayStation®Vita comprise two axes - horizontal and vertical - each with a value scope of 0x00 to 0xFF. The central value for both directions is 0x80. Note that there may be a margin of error up to approximately ±0x20; thus, it is necessary to set a dead zone of ±0x20 (ignore values 0x60 to 0xA0) in order to detect the neutral state. For details regarding the value scope of analog sticks, refer to the "Controller Service Overview" document. If the detection of the neutral state is not properly implemented on the application, this may cause problems such as, characters moving or the game view shifting, contrary to intended design.

The check mode for detecting such problems can be selected from this menu.

Check Target of Analog Sticks

Select the target for which to check the analog stick dead zone.

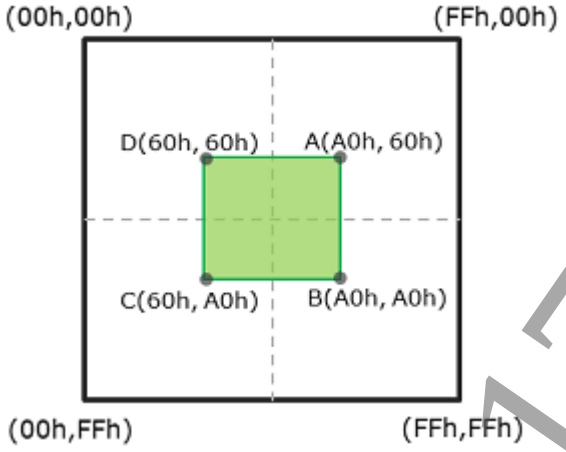
- **Off**: no check
- **Right Stick**: only check the right stick
- **Left Stick**: only check the left stick
- **Dual Sticks**: check both sticks

Check Mode

Select the check mode for checking the dead zone of analog sticks as follows. The selected mode will be applied immediately even while an application is running.

Deadzone Auto Walk

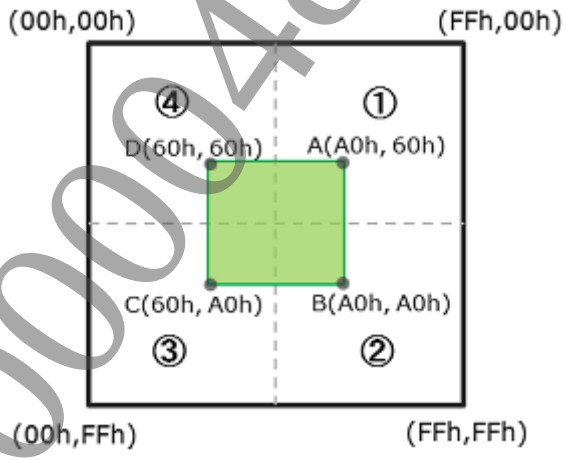
In this mode, a simulation is carried out where the values obtained by the analog stick transition from one edge of the dead zone to the other; use this mode to check the validity of software implementation within the dead zone.



When the output value of an analog stick within the dead zone (within ABCD of the above figure), the value obtained from the analog stick will transition in order from A to B, to C, and to D in two-second intervals. This operation will be canceled when the output value of the analog stick moves out of the dead zone. When the output value returns within the dead zone again, the transition operation will be resumed again from A to B, to C, and to D after two seconds.

Maximum Value Limitation

This mode should be used for developers to limit values to obtain from the analog stick to the edges of the dead zone and confirm by manual operation that center value processing is correctly carried out within the dead zone.



Mark off the output scope of the analog stick to areas 1 to 4 in the figure above and map output in each area to the corresponding edge of A, B, C, or D. The value that can be obtained from the analog stick will be A, B, C, or D in the above figure.

System - Region Settings

This menu item is possible to set a specific target territory for the DevKit/TestKit. Settable target territories are as follows:

- Japan
- North America
- Oceania
- UK
- Europe
- Korea
- Southeast Asia (including Hong Kong)
- Taiwan
- China
- Russia
- Mexico
- Off

To apply a target territory, it will be necessary to reboot the main unit. Also, when changing the target territory, settings such as display language, time zone, and button assignment will be reset to the values set at the time of shipment from factories in each target territory.

System - Restore Debug Settings

Restores the setting items that can be set with Debug Settings and Neighborhood for PlayStation®Vita for debugging purposes

Note
The setting items restored with this function can be checked on the screen displayed immediately prior to initialization.
Settings that are not included on the list on this screen will not be restored.

System - Show Information

This menu item displays system-related information.

Core Dump - Dump Level

This menu item allows selection of the dump level (content volume) when the core dump function outputs a core file from the two items listed below.

Mini Dump

This setting records the minimum required information and memory contents only. Compared to **Full Dump**, the output time is shorter and the file size following output is also smaller, but the output information amount being limited, information that cannot be referenced during debugging may exist. In particular, memory dump output is possible only for some areas.

Full Dump

This setting records all the information and memory contents that can be output with the core dump function. In particular, with regard to the process memory, this setting is effective for debugging because almost all the mapped memory areas are dumped, but the output time and the size following output tend to be larger compared to **Mini Dump**.

Mini Dump is set by default.

Core Dump - Uploader

The settings of the core file upload feature can be changed by changing the following items. By using the core file upload feature, the generated core file can be sent to a specified web server.

Enable Uploader

This item sets the core file upload feature to enabled/disabled.

- **Off**: enables the function.
- **On**: disables the function.

Off is set by default.

Uploader URL

This item specifies the URL of the web server for sending the core file. Up to 255 UTF-8 characters can be specified for the URL.

Specification example:

```
http://uploadserver.com/corefile_receiver.php
```

Auto Upload

By turning **Auto Upload** to **On**, the confirmation dialog and comment input dialog will not be displayed before sending, and the core file will be automatically sent after the core dump.

- **Off**: disables the function.
- **On**: enables the function.

Off is set by default.

For details on the core file upload feature, refer to the "Core File Upload Feature" chapter of the "Core Dump Overview" document.

Core Dump - Copy

This item copies a core file recorded to the internal memory card in the PTEL-2000 series to an external memory card. When a core file is selected and the **Copy** button is tapped, the selected core file will be copied to an external memory card. This menu item will not be displayed if an external memory card is not inserted or if the device is the PTEL-1000 series.

Core Dump - Delete

This item deletes a core file generated in the internal memory card of the PTEL-2000 series.

When a core file is selected and the **Delete** button is tapped, the selected core file will be deleted.

This menu item will not be displayed if device is the PTEL-1000 series.

Activation - Activate by Storage

This menu item updates the expiration date of the DevKit/TestKit. An expiration date is set for the DevKit/TestKit as an illegal use prevention measure. This item updates the expiration date using the activation file (file that describes the expiration date) on the memory card.

When this menu is selected, the activation file is searched on the memory card (ux0:data/activate) and the discovered activation file is applied to the DevKit/TestKit to update its expiration date.

Once the update has succeeded, the DevKit/TestKit restarts. Following restart, the new expiration date can be checked by calling Activation - Show Expiration Date.

For details on the activation file placement method, etc., refer to the "DevKit/TestKit Activation User's Guide" document.

Activation - Activate by Connecting PC

This menu item updates the expiration date of the DevKit/TestKit. This item obtains an activation file stored on a PC through CMA and then updates the expiration date of the DevKit/TestKit.

When this menu is selected, an activation file stored on a PC with a given file name ("My Documents\PS Vita\package\activate\vita_activation.afv") is obtained and the file is applied to the DevKit/TestKit to update its expiration date.

Once the update has succeeded, the DevKit/TestKit restarts.

For details on the activation file placement method, etc., refer to the "DevKit/TestKit Activation User's Guide" document.

Activation - Show Expiration Date

This menu item displays the expiration date of the DevKit/TestKit. An expiration date is set for the DevKit/TestKit as an illegal use prevention measure. This item displays the remaining time until the expiration date elapses.

Display examples:

- "5 day +10:00:00"
  This indicates that the expiration date is 5 days and 10 hours from now.
- "This DevKit/TestKit is expired. See "DevKit/TestKit Activation User's Guide"."
  This indicates that the expiration date of the Development Kit elapses. For the method to update the valid date, refer to the "DevKit/TestKit Activation User's Guide" document.
- "This DevKit/TestKit is not activated."
  This is displayed when the Development Kit is in shipping condition. Since the expiration date has elapsed, update the expiration date as mentioned above.
- "The backup battery has failed."
  The backup battery inside the DevKit may have been completely consumed, or a hardware failure may have occurred. Contact Private support on the PlayStation®Vita Developer Network website for assistance.
- "Cannot check expiration date. Please set date via Internet."
  The time is not set correctly in the TestKit. Select **Date & Time -> Date & Time Settings -> Set via Internet -> Set Now** to set the date and time.

Activation - Show Activation Key

This menu item displays the internal ID to be used when updating the expiration date of the DevKit/TestKit. Normally, this item need not be used by the user, but it is one of the information items to be notified to the manager during troubleshooting. For details, refer to the "DevKit/TestKit Activation User's Guide" document.

Game - Debug Info

This function displays useful information notification for development.

- **Off**: disables the function.
- **On**: enables the function.

Off is set by default.

When set to **On**, the following notification will be displayed.

Display of Information Obtained for Save Data Slot Parameters

The following will be displayed upon calling `sceAppUtilSaveDataSlotGetParam()` of the application utility library.

```
AppUtil:
Read Slot Status <slot id>
```

`<slot id>` represents the save data slot ID of the parameter obtainment target.

> **Note**
> This setting alone will not display notifications. **System - TRC Check Notifications** must be separately enabled.

Game - Fake No Memory Card

For an application, this function fakes a state in which a memory card is not inserted. Use this function when testing operation that changes depending on the existence of a memory card. The following settings are possible.

- **Off**: disables the function.
- **On**: checks the operation of VC-VC configuration application.

When this item is set to **On**, the operation to be performed when a memory card does not exist is faked if a function that requires a memory card is called from an application. Note that this function does not fake the access to savedata0: and addcont0:. Specifically, the operations of the following library functions are faked.

- Photo mount function (photo0:) and music mount function (music0:) of application utility library
- Photo export library
- Photo Import Dialog library

When using the above described functions, check the operation to be performed when a memory card does not exist by setting this item to **On**.

Game - Fake Free Space (FS)

When the application performs writing to the save data area, this function fakes file system free space as 0 KB. Use this function when testing operation in case file system free space is insufficient when the application performs writing to the save data area.

The following settings are possible.

- **Off**: disables the function.
- **On**: enables the function.

If this setting is **On**, file system free space where the application's save data is mounted will be faked as 0 KB. Bear in mind that this function does not fake free space in such a manner as to guarantee that writing to the save data area will fail. Operations that would succeed with 0 KB of free space (such as overwrites, which do not cause insufficient free space) will succeed even if this function is set to **On**.

> **Note**
> This function does not fake save data quota. In order to fake save data quota, use **Fake Free Space (Quota)**. For file system free space, save data quota, and save data overwrites that do not cause insufficient free space, refer to the "Save Data User's Guide" document.

Game - Fake Free Space (Quota)

When the application performs writing to the save data area, this function fakes save data virtual free space as 0 KB. Use this function when testing operation in case remaining space for the save data quota is insufficient when the application performs writing to the save data area.

The following settings are possible.

- **Off**: disables the function.
- **On**: enables the function.

If this setting is **On**, save data virtual free space in the application's save data area will be faked as 0 KB. Bear in mind that this function does not fake free space in such a manner as to guarantee that writing to the save data area will fail. Operations that would succeed with 0 KB of save data virtual free space (such as overwrites, which do not cause insufficient free space) will succeed even if this function is set to **On**.

> **Note**
> This function does not fake file system free space. In order to fake file system free space, use **Fake Free Space (FS)**.
> For file system free space, save data quota and save data overwrites that do not cause insufficient free space, refer to the "Save Data User's Guide" document.

Game - Fake Contents Max

This function fakes the behavior to be performed when the number of contents that can be registered to the system from within an application reaches the upper limit. In the case that an application calls the contents registration function, use this function to test the operation to be performed when the registration fails for having reached the maximum number of contents. The following settings are possible.

- **Off**: disables the function.
- **On**: enables the function.

When this item is set to **On**, the call of the following library from within an application will result in a failure with an error for the excessive number of contents.

- Photo export library

When using the above described functions, check the operation in the case that the number of contents reaches the upper limit by setting this item to **On**.

Game - Fake Save Data Slot Broken

This function makes an arbitrary save data slot appear corrupted.

- **Off**: disables the function.
- **On**: <slot id>
  : enables the function. (The save data slot ID from 000 to 255, entered in the following manner, will be displayed in slot id.)

Off is set by default.

When set to **On**, the on-screen keyboard is displayed for entering the save data slot ID to appear as corrupt. The save data slot of the specified ID will always be in the corrupted state. Only 1 slot can be set to be in the corrupt state.

This function can be used to check the application's behavior when a slot is corrupted. Because changes made to this setting are immediately reflected, corruption check of the save data slot can be performed at arbitrary points in the application by suspending the application and enabling this function.

> **Note**
> Reference information obtained regarding save data slot parameters (displayed by enabling the **Game - Debug Info** setting) to confirm which slot to corrupt.
>
> This function will not affect the ★ **Save Data Slot** display. For ★ **Save Data Slot**, refer to Chapter 7 "Development Support Features by Application".

Game - Fake Trophy Earning

Regardless of the account of the current user, the application will always be started with rights for obtaining trophies available. The information on rights for obtaining trophies is recorded on the save data. However, in case you are using save data across multiple environments during development, use this function for testing the operation of trophy acquisition.

The following settings are possible.

- **Off**: disables the function.
- **On**: enables the function.

Trophy acquisition will always succeed for applications started up with this setting **On**. The setting will remain effective until the application is terminated.

The restrictions set by an application independently will not occur by using the following APIs of NP Title User Storage library and NP ScoreRanking library only when the development environment or QA environment is used.

- `sceNpTusChangeModeForOtherSaveDataOwners()`
- `sceNpScoreChangeModeForOtherSaveDataOwners()`

During development, it is possible to perform the operation test without changing the code of the application in which these APIs are incorporated when loading save data from a different environment.

Game - Init Safe Memory

This menu item is a function for forcibly initializing safe memory at application start-up. Use it for testing operation when safe memory is initialized.

The following settings are possible.

- **Off**: disables the function.
- **On**: enables the function.

If the application is started up with this setting **On**, safe memory will be initialized.

> **Note**
> Safe memory is initialized when power is turned off forcibly by keeping the **power** button pressed during application runtime, or by removing the PlayStation®Vita card/memory card that is the save destination of save data (abnormal-type operation), but the removal of PlayStation®Vita card/memory card in particular risks damaging the file system. Therefore, for safety purposes use this setting to initialize safe memory.

Game - Add-On Data (PS Vita)

It is possible to display information on add-on data for PlayStation®Vita/PlayStation®TV saved on memory cards and PlayStation®Vita cards and carry out the following.

- Individually disable (enable) right information of add-on data by selecting each add-on data on the list. This can be used to test error handling in case add-on data rights are lost. Add-on data without usage right will have its title name displayed with a line across it.
- When a right to add-on data is not owned, the function for mounting add-on data from within the application `sceAppUtilDrmOpen()` will return an error indicating the absence of rights.
- The entire title names of add-on data can be confirmed. A button will be displayed to show the entire title names on a separate page.

Graphics Library (libgxm) - Use Debug Version of libgxm for Game

This is a debug function related to libgxm. It sets whether the debug version of libgxm is used during game development. The settings can be selected from **Enable** and **Disable**. When **Enable** is selected, the debug version of libgxm is used.

Content Downloader

This menu item is a function for downloading and installing the game packages from the Web server on the network. Use it, for example, when installing the latest package for QA or other purposes.

With the first input screen, input URL of the directory where the package files are stored. (In order to use this function the directory index function should be enabled on the Web server side beforehand.) Tapping **OK** after inputting URL displays the list of packages stored in the Web server. Then, mark the packages you wish to download and install from among the list. Tapping **Download** button starts the download processing in the background and returns to the initial screen. To check the progress of the download and installation processing, return to the home screen using **PS** button and check the notification indicator located on the upper right of the screen.

Debug Settings Usable With Each Device for Development and Boot Parameter

The usable settings vary depending on the device (DevKits and TestKits) used in development and DevKit boot parameter settings (**PS TV Emulation** and **Release Check Mode**). A summary of the usable settings for these devices and each of the boot parameters is as follows.

Setting Items		DevKit				TestKit	
		PS TV Emulation				PTEL-1000 Series	PTEL-2000 Series
		Off		On			
		Development Mode	Release Mode	Development Mode	Release Mode		
Boot Parameters		Yes	Yes	Yes	Yes		
	Release Check Mode	Yes	Yes	Yes	Yes		
	Release Mode Console		Yes		Yes		
	Memory Size	Yes		Yes			
	PS TV Emulation	Yes	Yes	Yes	Yes		
System Update		Yes	Yes	Yes	Yes	Yes	Yes
	Update Server URL	Yes	Yes	Yes	Yes	Yes	Yes
	Show PUP List	Yes	Yes	Yes	Yes	Yes	Yes
	Enable Host0	Yes		Yes			

Setting Items	DevKit				TestKit	
	PS TV Emulation					
	Off		On		PTEL-1000 Series	PTEL-2000 Series
	Development Mode	Release Mode	Development Mode	Release Mode		
Network	Yes	Yes	Yes	Yes	Yes	Yes
Network Emulation	Yes	Yes	Yes	Yes	Yes	Yes
Intermittent Connection	Yes		Yes			
Fake 3G Interface	Yes	Yes			Yes	Yes
NAT Traversal Information	Yes	Yes	Yes	Yes	Yes	Yes
PSP Adhoc SSID prefix	Yes	Yes	Yes	Yes	Yes	Yes
PSN℠	Yes	Yes	Yes	Yes	Yes	Yes
Account Selection	Yes	Yes	Yes	Yes	Yes	Yes
Account Name	Yes	Yes	Yes	Yes	Yes	Yes
NP Environment	Yes	Yes	Yes	Yes	Yes	Yes
Fake Plus	Yes	Yes	Yes	Yes	Yes	Yes
NP Debug	Yes	Yes	Yes	Yes	Yes	Yes
In-Game Commerce Debug	Yes	Yes	Yes	Yes	Yes	Yes
Service ID	Yes	Yes	Yes	Yes	Yes	Yes
Upgradable App Debug	Yes	Yes	Yes	Yes	Yes	Yes
Patch Test	Yes	Yes	Yes	Yes	Yes	Yes
TPPS Proxy	Yes	Yes	Yes	Yes	Yes	Yes
Trophy Setup Dialog Debug	Yes	Yes	Yes	Yes	Yes	Yes
Web API Rate Limit Threshold	Yes	Yes	Yes	Yes	Yes	Yes
Ignore NpTitleId set by API in Development Mode	Yes		Yes			
Location Data	Yes	Yes			Yes	Yes
Emulate Permission	Yes	Yes			Yes	Yes
Permission Status	Yes	Yes			Yes	Yes
Device Model	Yes	Yes			Yes	Yes
Make AP Invisible	Yes	Yes			Yes	Yes
System	Yes	Yes	Yes	Yes	Yes	Yes
○ Button Behavior	Yes	Yes	Yes	Yes	Yes	Yes
Slow Card Mode	Yes	Yes	Yes	Yes	Yes	Yes
TRC Check Notifications	Yes	Yes	Yes	Yes	Yes	Yes
GPI Switch	Yes		Yes			
HDMI Audio Output	Yes					
HDMI RGB Full Range	Yes					
PRX Debug	Yes		Yes			
Debug Network Clock	Yes	Yes	Yes	Yes	Yes	Yes
Reset Network Clock	Yes	Yes	Yes	Yes	Yes	Yes
Deadzone Check of Analog Sticks	Yes	Yes			Yes	Yes
Region Settings	Yes	Yes	Yes	Yes	Yes	Yes
Restore Debug Settings	Yes	Yes	Yes	Yes	Yes	Yes
Show Information	Yes	Yes			Yes	Yes
Core Dump	Yes	Yes	Yes	Yes	Yes	Yes
Dump Level	Yes	Yes	Yes	Yes	Yes	Yes
Uploader	Yes	Yes	Yes	Yes	Yes	Yes
Copy						Yes
Delete						Yes

Setting Items	DevKit				TestKit	
	PS TV Emulation					
	Off		On		PTEL-1000 Series	PTEL-2000 Series
	Development Mode	Release Mode	Development Mode	Release Mode		
Activation	Yes	Yes	Yes	Yes	Yes	Yes
Activate by Storage	Yes		Yes		Yes	Yes
Activate by Connecting PC	Yes		Yes		Yes	Yes
Show Expiration Date	Yes	Yes	Yes	Yes	Yes	Yes
Show Activation Key	Yes	Yes	Yes	Yes	Yes	Yes
Game	Yes	Yes	Yes	Yes	Yes	Yes
Debug Info	Yes	Yes	Yes	Yes	Yes	Yes
Fake No Memory Card	Yes	Yes	Yes	Yes	Yes	Yes
Fake Free Space (FS)	Yes	Yes	Yes	Yes	Yes	Yes
Fake Free Space (Quota)	Yes	Yes	Yes	Yes	Yes	Yes
Fake Contents Max	Yes	Yes	Yes	Yes	Yes	Yes
Fake Save Data Slot Broken	Yes	Yes	Yes	Yes	Yes	Yes
Fake Trophy Earning	Yes	Yes	Yes	Yes	Yes	Yes
Init Safe Memory	Yes	Yes	Yes	Yes	Yes	Yes
Add-On Data (PS Vita)	Yes	Yes	Yes	Yes	Yes	Yes
Graphics Library (libgxm)	Yes		Yes			
Use Debug Version of libgxm for Game	Yes		Yes			
Content Downloader	Yes	Yes	Yes	Yes	Yes	Yes

5 Functions of Application

Media Contents Distribution - Host Link Features

Memory card is used as the media contents distribution destination in PlayStation®Vita.

Memory card is required in order to use Photos / Music / Videos applications or install packaged games.

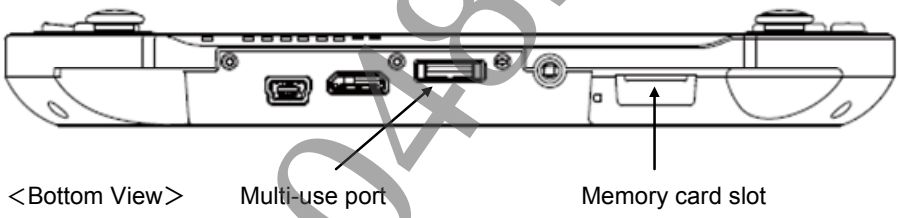
In the case of the DevKit/TestKit, the system must be started up with the memory card inserted in the memory card slot at the bottom right of the main unit. The memory card can be inserted or removed, but make sure not to insert or remove it while the power is on.

To delete the data on the memory card, execute **Format -> Format Memory Card** in the Setting application.

Following update to the system software 0.990 or later and prior to using all the host link features, be sure to select **Settings -> Format -> Restore This System** to initialize PlayStation®Vita.

To place each media content of Photos/Music/Videos application in the memory card, first install Content Manager Assistant for PlayStation®Vita DevKit (hereafter, CMA) on the Windows PC, and then connect the USB cable that is bundled with the DevKit/TestKit to the Windows PC that is to be the host (the cable differs from the USB cable for debugging connected to the USB Mini-B port for development). For the CMA usage method, refer to the "Content Manager Assistant Overview" document. The CMA being a different module than the SDK package, download this package via SDK Manager.

Figure 8 DevKit Memory Card Slot and Multi-use Port Locations



Once the connection of PlayStation®Vita and CMA has been recognized, the Content Manager application icon on the home screen of PlayStation®Vita blinks. The icon returns to its normal state when Content Manager application is started up or the connection is canceled. Applications can also be started up according to one's aim from Content Manager application.

If you become unable to transfer media contents or cannot access (play back, delete, etc.) contents, execute **Format Memory Card** from **Format** of the Setting application, and then transfer (access) the contents again. If this fails to solve the problem, execute **Restore This System** from **Settings** and then try contents transfer (access) again.

Music Application

To start up Music application, the memory card must be inserted in the DevKit/TestKit. To play back music, the music contents must be transferred from the Windows PC, referring to the above-described media contents placement method.

The Music application start-up procedure is described below.

(1) Place the contents on the DevKit/TestKit from the Windows PC

In the default state, the music contents of the types that can be played back with PlayStation®Vita are placed in the "My Music" folder on the Windows PC.

The playable formats are MP3, AAC, and WAV. (The .mp3, .m4a, .mp4, .aac, and .wav extensions are supported.)

(2) Install CMA for PSP2 DevKit to the Windows PC

(3) Start up the DevKit/TestKit

Insert the memory card before starting up the DevKit/TestKit.

(4) Connect the PC to which CMA have been installed to the DevKit/TestKit with the USB cable.

(5) Tap the "Music" icon on the home screen and tap the gate at the center of the displayed LiveArea™

Music application starts up.

When the CMA is recognized, the "PC" list is displayed on the top menu of the Music application, so transfer the music contents on the Windows PC.

(6) Once the Music application screen is displayed, tap "PC"

Browse the contents on the PC. Display the contents to be transferred, tap the **option** button at the bottom right, tap the copy menu, select the file to be transferred, and then tap **Copy**.

Photos Application

Below is an explanation of the features and usage method of the Photos application.

In order to start up the Photos application, the memory card must be inserted in the DevKit/TestKit.

The memory card must be inserted prior to starting up the DevKit/TestKit, and must never be removed and inserted back again while the DevKit/TestKit, is running.

Photos Capture Feature

The Photos application provides a feature to manage still images and movies (photo contents).

- Photo Contents List Display
  It is possible to display a list of photo contents imported to the system software.
- Display a Single Photo of Photo Contents
  You can view a photo full-screen by selecting 1 of photo contents on the list.
- Delete Photo Contents
  It is possible to delete photo contents displayed on the list or full-screen.
- Camera Shot
  The camera shot mode enables camera shot of still images and movies with either the front or back camera.

Friends Application

The explanations of the Friends application are described below.

Sony Entertainment Network Account

A Sony Entertainment Network account is necessary to use the Friends application.

Friends Management Features

The Friends application provides a number of management features for friends.

- Send Friend Registration Request
  Enter an online ID and send a friend registration request.
- Accept/Refuse Friend Registration Request
  Acceptance/refusal from the profile of a player from whom a friend registration request has been received is possible.
  Acceptance can also be done from the list of players from whom a friend registration request has been received.
- Delete Friend
  The profiles of friends on the friend list can be displayed and deleted.
- Cancel Friend Registration Request
  Cancellation (deletion) from the profile screen of a player who has made a friend registration request is possible.

Blocked User List Management Features

The Friends application provides a number of blocked user list management features.

- Register to Blocked User List
  It is possible to remove players from the friend list and to block messages from given players.
- Delete from Blocked User List
  Release the block on blocked players.

Precautions

If the friends list is not correctly displayed, restore the PlayStation®Vita system.

Internet Browser Application

Below is a description of the features and usage method of the Internet browser.

Internet Browser Features

Start-up

The Internet Browser is mainly started up with the following methods:

- Tapping the icon on the home screen
  This is the regular start-up method. It restores window state as it was when last terminated.
- Tapping on another application's link on LiveArea™
  This creates a new window, opening the specified webpage.
- Tapping another application's keyword search icon on LiveArea™
  This creates a new window, executing an Internet search with the specified keyword.
- Tapping on an online manual icon
  This creates a new window, opening the online manual page of the specified application.
- Start-up from another application
  The browser is started up in response to a call from another application as an application associated with an "http:" or "https:" scheme.
  A new window will be created, opening the specified webpage.

Webpage Display

The Internet Browser has the following features for displaying webpages:

- Address input
  Displays the webpage with the URL input by the user.
- Internet search
  Displays the Internet search result page obtained from the keyword input by the user.
- Selection from bookmarks/history
  Displays a webpage from a bookmark item or history item selected by the user.
- Start-up via URL specification
  Displays the webpage with the URL specified at start-up

Content Operation

The Internet Browser enables the following operations on the contents (webpages) displayed:

- Going back/going forward/stopping/reloading
- Scrolling
- Zooming in/zooming out/fitting to the screen
- Inputting forms
- Saving images
- Registering bookmarks

Downloading

For some formats, the Internet Browser supports downloading contents such as music, photos and videos that the PlayStation®Vita can reproduce.

- Music
  AAC, MP3, MP4, RIFF, WAV
- Photo
  BMP, GIF, JPEG, PNG, TIFF
- Video
  MP4, MNV, MGV

Bookmark and History Management

The Internet Browser has bookmark and history management features.

- Bookmark registration
  You can register the webpage that is being displayed, or an arbitrary webpage as a bookmark. A maximum of 1000 bookmarks can be registered.
- Bookmark editing
  You can delete, sort or edit the contents of bookmark items.
- Folders
  You can arrange bookmark items in folders of up to three layers.
  You can create up to 100 folders.
- Preset bookmarks
  Bookmarks contain bookmark items registered in advance in the system.
  These cannot be edited/deleted.
- History registration
  Basically, all webpages displayed by the user are automatically registered in the history.
  History is always displayed in chronological order (from the newest to the oldest).
- History deletion
  History items can only be deleted en bloc. Items cannot be deleted individually.
  Also, history items older than 6 days are deleted automatically.

Multi-Window Management

The Internet Browser supports multiple windows, and has a feature to manage a maximum of eight windows.

- Window creation
  New windows are created in response to user operation and requests from the contents.
  If the number of windows exceeds 8, the oldest windows will be deleted automatically.
- Window deletion
  Windows are deleted in response to user operation and requests from the contents.
  If the number of windows reaches 0, a new window will be created automatically.
- Saving window state
  Even if the application is terminated, the last state (the number of windows and the URLs of the webpages that were opened in each of them) will be saved, and restored at the next start-up.

Indicators

The Internet Browser displays the indicators below:

- Loading progress
  While loading a webpage, progress will be displayed on a blue bar on the address bar.
- SSL Icon
  When an SSL-protected webpage is being displayed, the SSL icon is displayed on the address bar.

How to Use the Internet Browser

Start-up

When the Internet Browser is started up by tapping the icon on the home screen, the Internet Browser's LiveArea™ will be displayed.

On LiveArea™, there are a gate and a window list. Tap on either of these to switch to full screen.

- Tapping on the gate
  This will activate the window that was last active, and display its webpage.
- Tapping on an arbitrary item on the window list
  This will activate the selected window, and display its webpage.

Webpage Display

You can display webpages with one of the following methods:

- Address input
  Edit the URL by tapping on the address bar.
- Internet search
  Tap the Internet search icon on the toolbar to enable search keyword input.
- Bookmark/history selection
  Tap on a bookmark icon on the toolbar to open a bookmark.
  History is integrated in the bookmarks as the History folder.

Content Operation

You can operate the contents with the following methods:

- Going back/going forward/stopping/reloading
  Tap the **back/forward** icon on the toolbar, or the **stop/reload** button in the address bar.
- Scrolling
  Drag or flick the contents.
- Zooming in/zooming out/fitting to the screen
  Zoom in by pinch out and zoom out by pinch in.
  Also, double-tap to zoom so as to fit the area containing that spot to the width of the screen.

- Inputting forms
  By tapping on the relevant part of a form, various input methods will be displayed as necessary.
- Saving images
  Select **Save Image** from the pop-up menu displayed by tapping and holding an image displayed in the contents.
- Registering bookmarks
  Select **Add Bookmark** from the menu displayed by tapping on the **option** button.

Downloading

You can download contents with the following method:

(1) Tap on the link, etc. for referencing the relevant contents
(2) The system determines whether the contents are downloadable
(3) If the contents are downloadable, the download confirmation screen is displayed
(4) Tap **Save** on the download confirmation screen

Bookmarks and History

Tap on the bookmark icon on the toolbar to display the bookmark screen.

You can manage bookmarks and history on that screen in the following ways:

- Editing bookmarks
  Tap the **option** button on the bookmark screen, and select **Edit** on the menu that is displayed.
- Deleting bookmarks
  Tap the **option** button on the bookmark screen, and select **Delete** on the menu that is displayed.
- Creating folders
  Tap the **option** button on the bookmark screen, and select **Create Folder** on the menu that is displayed.
- Sorting
  Tap the **option** button on the bookmark screen, and select **Sort** on the menu that is displayed.
- Clearing history
  Go to the History folder on the bookmark screen, and select **Clear History** on the menu that is displayed by tapping the **option** button.

Multi-Window Management

Tap the window icon on the toolbar to display the window screen.

You can manage windows on that screen in following ways:

- Creating windows
  Tap the button for adding windows on the window screen.
- Deleting windows
  Tap the **deletion** button on the upper left of the relevant window on the window screen.

PS3 Remote Play Application

How to use the PS3 Remote Play application is explained below.

Register the device to PlayStation®3

Device registration to PlayStation®3 is required in order to use PS3 Remote Play. In order to register the PlayStation®Vita, use version 4.00 or later of the PlayStation®3 system software.

From the system software of PlayStation®3, select **Settings -> Remote Play Settings -> Register Device -> PS Vita** to display the value on the screen. Next, start the PS3 Remote Play application from PlayStation®Vita and input the value displayed on the screen of PlayStation®3. The device registration procedure is completed when the correct value has been input.

Connect to PlayStation®3

Select **Network -> Remote Play** from the system software of the PlayStation®3 for which device registration was performed. Then start the PS3 Remote Play application of PlayStation®Vita and select the connection path to start connection to PlayStation®3. Upon completion of the connection, the screen of PlayStation®3 is displayed on PlayStation®Vita.

Following connection completion, PlayStation®3 can be operated using the buttons and analog sticks of PlayStation®Vita.

Disconnect from PlayStation®3

Press the **PS** button of PlayStation®Vita to quit the PS3 Remote Play application. This cuts off the communication with PlayStation®3.

Cross-Controller Application

The Cross-Controller application is explained in this section. The Cross-Controller application is started by tapping the icon displayed in the LiveArea™ of the PS3 Remote play application.

To use the Cross-Controller application, the system utility of the Cross-Controller must be called from the PlayStation®3 side.

Download Feature

This feature first searches for, and detects, a PlayStation®3 on the local network. The PlayStation®Vita application package to coordinate with the PlayStation®3 application is downloaded to PlayStation®Vita. The application will automatically start installing when it is downloaded and the screen returns to the home screen.

Launcher Feature

This feature searches for a PlayStation®3 on the local network and starts the application installed on PlayStation®Vita to run in coordination with the application running on the detected PlayStation®3.

It is also possible to not use this feature and start the application registered to the home screen and directly let it run in coordination with a PlayStation®3 application.

PS4 Link Application

This explains the PS4 Link application.

The PS4 Link application includes two features used when a PlayStation®Vita and PlayStation®4 are connected: a PS4 remote play feature and a second screen feature.

In particular, the DevKit/TestKit remote play feature provides feature such as for checking the key assignment for debugging. For details on development with these features, refer to the "Remoteplay Library Overview" and "Companion Application System Overview" documents distributed as SDK documents for PlayStation®4.

Save Destination of Application Data Relating to Accounts

Application data includes data relating to the accounts, which is stored in separate locations by account entry.

While retail units only have 1 account entry, the DevKit/TestKit allows creating up to 64. In particular, with regard to data on the memory card (ux0:), note that, when using a memory card on multiple DevKits/TestKits, referencing the intended data will not be possible if the account entry is different from that of the original DevKit/TestKit.

For the account entry, refer to the "PSN℠ - Account Selection" section in Chapter 4 "★Debug Settings Functions".

Below is a list of data relating to the accounts:

Save data	: data and safe memory contents stored in savedata0: by the application
Trophy	: set-up information and acquisition information
Friends	: friend list information and play history information
Messages	: cache information of received messages
"near"	: collected information, play history
Activity	: cache information

Note
LiveArea™ information will not change individually when switching accounts. For this reason, it is possible that the contents displayed with other accounts and those displayed with the account after switching may get mixed up. You can initialize LiveArea™ information by switching language settings.

6 Safe Mode Features

Safe mode is an operation to boot up the DevKit/TestKit with the minimum necessary features in case the system cannot boot up normally. By performing boot-up in safe mode and executing the menu, improvement of symptoms may be possible. Although it will not always improve symptoms, try booting up in safe mode if the system cannot boot up normally.

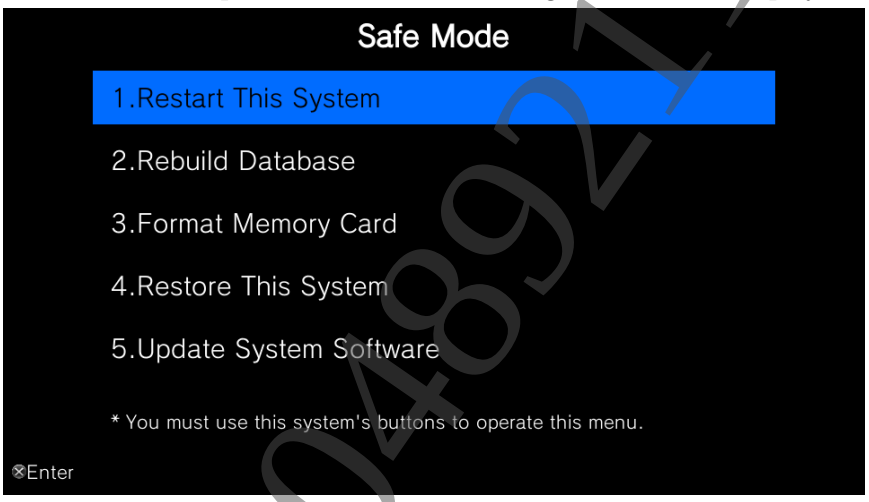
Safe Mode boot-Up Procedure

Follow the procedure below to boot up in safe mode:

(1) Power off the DevKit/TestKit.

(2) Boot up by pressing the buttons of the DevKit/TestKit
Keep the **PS** button + **power** button + **R** button pressed simultaneously for over 5 seconds (press for longer than 5 seconds with watching the console display of the development host computer). The system will boot up automatically, with the application running in safe mode.

(3) When safe mode boot-up is successful, the following screen will be displayed:



Safe Mode Features and Influence of Their Execution

If the safe mode boot-up is successful, try the following steps for improvement:

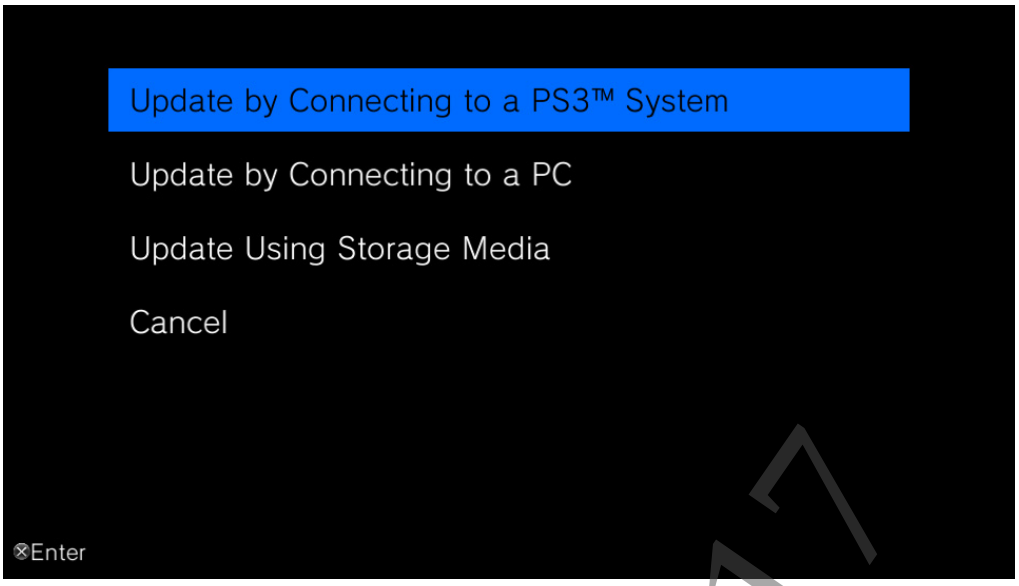
(1) Restart This System
Terminate safe mode and perform normal boot-up.

(2) Rebuild Database
Rebuild the database storing the application information displayed on home screen at the time of normal boot-up.

(3) Format Memory Card
Processing is executed as described in "Format Memory Card" of Chapter 3 "Setting Functions".

(4) Restore This System
Processing is executed as described in "Restore This System" of Chapter 3 "Setting Functions".

(5) Update System Software
The system software is updated. On the following screen, select one of the followings: **Updated by Connecting to a PS3™ System**, **Update by Connecting to a PC** or **Update Using Storage Media**.



Procedure for Executing System Update in Safe Mode

When booting up in safe mode, follow the procedure below for system update:

(1) Prepare system update files

A system update file is installed in the following directory:

```
%SCE_ROOT_DIR%\PSP2 SDKs\System Update Files\PSP2UPDAT-<VERSION>.PUP
```

When performing update by storing the system update file on the development host computer [only for DevKit]

Place the system update file so that they may be referenced with the path below and select **Update Using Storage Media**. Note that the update data file name must be changed.

For `host0:`, it is possible to specify a given directory on the development host computer side with the "`psp2ctrl fsroot <DIR>`" command or with Neighborhood for PlayStation®Vita.

```
host0:PSP2/UPDATE/PSP2UPDAT.PUP
```

Note
Note that when the update data has been placed on `host0:`, **Update Using Storage Media** must be selected instead of **Update by Connecting to a PC**.

Note
The connection between the DevKit and the development host computer must be established with the `psp2ctrl connect` command.

When performing update by storing the system update file on the memory card

Place update data files in ux0:data/PSP2/UPDATE/PSP2UPDAT.PUP using the `psp2ctrl` command, and select **Update Using Storage Media.**

```
> psp2ctrl cp PSP2UPDAT.PUP ux0:data/PSP2/UPDATE
```

Use DevKit to write PUP into the memory card.

When performing Update by Connecting to a PC by storing the system update file on the HTTP server

Set up beforehand the HTTP server and the update list file according to the procedure described in the "System Update" section in the "DevKit/TestKit Setup Guide" document, and set up CMA on the PC.

The URL for referencing the update data list on the HTTP server from the PC is described in plain text in the following file.

```
host0:PSP2/UPDATE/SERVER_URL.TXT
```

When this file cannot be found, the error "One or more settings are not valid." will be displayed.

When performing Update by Connecting to PlayStation®3 by storing the system update file on the HTTP server

Set up beforehand the HTTP server and the update list file according to the procedure described in the "System Update" section in the "DevKit/TestKit Setup Guide" document, and use version 4.00 or later for the system software of the PlayStation®3.

The URL for referencing the update data list on the HTTP server from the PlayStation®3 is described in plain text in the following file.

```
host0:PSP2/UPDATE/SERVER_URL.TXT
```

When this file cannot be found, the error "One or more settings are not valid." will be displayed.

(2) Display console output [only for DevKit]

Run the command prompt and input the following command:

```
> psp2ctrl console
```

Alternatively, display **Console Output** of Neighborhood for PlayStation®Vita.

(3) Select Update System Software from the safe mode menu

Begin system update by following the wizard and inputting "Yes" in response to the confirmation dialog "Perform system update? ". The system will reboot automatically once system update is complete.

7 Development Support Features by Application

Individual development support features are available for applications under development.

Note
Use ★**Debug Settings** for common support features for all applications.

Precautions

Basically, only English is supported for text such as setting item names and options in the development support features.

Development Information Display Features

Follow the procedure below to transition to the **Information** screen:

(1) Tap and hold the home screen and transition to the edit mode screen.

(2) On the edit mode screen, tap the icon of the application under development installed as a package.

(3) Tap **Information** on the menu that is displayed.

In the PlayStation®Vita, this screen is used by users to look up various application information. In the DevKit/TestKit, the information displayed is more extensive.

★ Title ID

Displays the application's product code.

★ PS Vita card R/W Area xxxxMB/yyyyMB

If the PlayStation®Vita card is inserted, the size of writable area in the PlayStation®Vita card is displayed. xxxx and yyyy represent the current usage amount and the total amount respectively.

★ Upgradable App xxxx

Displays the state of upgradable application that has been installed.

Setting Value	SKU Flag Value
Trial	Trial version supporting the upgradable application
Full	Full version supporting the upgradable application
Off	Full version not supporting the upgradable application

★ Save Data Account xxxxx

Displays save data account information. xxxxx will be one of the following:

Online

This is save data created while the main unit is signed up. The DevKit/TestKit allows switching Sony Entertainment Network accounts; however, note that if save data is created with an entry that is not signed up, it will not become Online.

Local

This is save data created while the main unit is not signed up. When the main unit is signed up and the application is started up while in this state, save data will automatically transition to Online (in some cases, it may transition without start-up). However, the application must be started up by a user with rights for earning trophies.

Note
Rights for obtaining trophies are displayed under the **Trophy Earning** item of the **Information** screen. This information will be displayed not only on the DevKit/TestKit but also on retail units. If **Trophy Earning** is **Eligible**, it means that the owner of the save data and the owner of the main unit are found to match.

★ Save Data Quota xxxxKB/yyyyKB

Displays save data size. xxxx is the space that is currently used, while yyyy is the total space.

This value can be used to check the save data size reported when submitting the master. The xxxx value is reported as the "Minimum save data size required to run the application (also to be written on a package)." Note that yyyy is the value set in Save Data Quota of the application's param.sfo.

If there are no save data yet for that application, xxxx and yyyy will not be displayed. An application's save data is created when the application is first started up. Also, since xxxx includes the size of the system directory created by the system, 768 KB will be consumed even if the application has not stored any save data.

Note
In order to enable this display, the application's save data must be properly created and placed as follows.

```
ux0:user/<user id>/savedata/<title id>/
```

For save data to be created in the proper place, the application package must be installed and started up. However, note there is a possibility that the directory for development (as exemplified below) is referenced as save data during development and save data may not be created in the proper place.

```
host0:savedata/
ux0:data/savedata/
```

For details on where save data is created, refer to the "Application Development Process Overview" document.

★ Save Data Slot xxx...

The save data slot information saved in the save data is displayed. The slot number created is listed in "xxx..." If the save data slot has a broken status at this time, "*" is displayed following the slot number.

Example) ★ Save Data Slot 0, 1, 2*, 3

Application Testing Feature

Follow the procedure below to transition to the ★ **Check** screen:

(1) Tap and hold the home screen and transition to the edit mode screen.
(2) On the edit mode screen, tap the icon of the application under development installed as a package.
(3) Tap ★ **Debug Utility** on the menu that is displayed.
(4) Tap ★ **Check** on the menu that is displayed next.

★ Check

Collectively tests applications, patches and additional contents, and displays the results of testing. The results of testing will be stored in text file format under ux0:/data/check.

Known Limitations

When ★ **Check** is executed for a suspended application, the message below will be displayed, and testing will not be possible; perform testing after terminating the application.

- Check module initialization failed. (Application is in use or not found)

Individual Content Deletion Features

Follow the procedure below to transition to the menu screen:

(1) Tap and hold the home screen and transition to the edit mode screen.

(2) On the edit mode screen, tap the icon of the application under development installed as a package.

(3) Tap ★ **Debug Utility** on the menu that is displayed.

(4) Tap any items on the menu that are displayed next.

★ Delete Savedata

Only deletes the application's save data. Note that save data shared with other applications will also be deleted.

Targets only the save data of the account entry for the current user.

★ Delete All Savedata

Only deletes the application's save data. Note that save data shared with other applications will also be deleted.

Targets all save data of account entries for all users.

★ Delete Patch

Only deletes the application's patch data. Patch data cannot be shared with other applications.

★ Delete AddCont

Only deletes the application's additional content data. Note that additional content data shared with other applications will also be deleted.

Core File Generation Feature

Follow the procedure below to transition to the menu screen:

(1) Tap and hold the home screen and transition to the edit mode screen.

(2) On the edit mode screen, tap the icon of the application under development installed as a package.

(3) Tap ★ **Debug Utility** on the menu that is displayed.

(4) Tap ★ **Generate Core File** on the menu that is displayed next.

★ Generate Core File

This generates a core file for suspended applications. If core file generation is successful, the file path of the generated core file is displayed. An error message is displayed if generation is not possible. The core files generated here are equivalent to the core files described in the "Core Dump Overview" document.

8 Owner Identification When Inserting Memory Cards

When a memory card is inserted into the PlayStation®Vita main unit, the system checks whether the owners of the memory card and of the main unit match. If it finds that the memory card belongs to a different owner than the main unit, the system will request the user to initialize the memory card. The user can refuse initializing the memory card, but in this case, the memory card will not be recognized even if it is left inserted.

Writing Information to a Memory Card

When a memory card is used with the main unit, the following information will be written automatically:

- The main unit's device ID
- The main unit's account ID if it is signed up to Sony Entertainment Network

If the memory card is inserted into another main unit, or if it is removed and reinserted into the same main unit, the system will use this information to check whether the memory card's owner matches that of the main unit.

How the Owner Is Identified

The system identifies the owner based on main unit-memory card combinations with the following conditions:

System	Memory Card	
	Has No Account ID	Has an Account ID
Is Not Signed Up	Owner deemed identical (due to lack of information for identification)	Different owner (prompts main unit sign-up)
Is Signed Up	Same device ID: same owner	Same account ID: same owner
	Other device ID: same owner (checks whether it is the memory card's owner)	Other account ID: different owner

When the Owner Is Found to Be the Same

The memory card can be used without initializing it.

However, initialization may be necessary depending on the options in cases where the user confirmation is requested.

Even if the memory card's owner is found to be the same, the memory card's contents also have rights information. Cases in which the owner of the memory card and the owner of content rights do not match are rare; however, users without rights may not be able to start up the application or to earn trophies.

When the Owner Is Found to Be Different

The memory card cannot be used as it is. The user will need to perform initialization to use it.

However, in cases where use is possible by signing up the main unit, sign up will first be urged. In this case, if it is possible to sign up with the same account ID as the memory card, the owners will be found to match.

Identification With the DevKit/TestKit

In order to ensure smooth application development, in the DevKit/TestKit memory card owner identification is not performed, and owners are always treated as matching. In addition, writing of information onto the memory card is not performed either.