Document List

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Table of Contents

1 About This Document

This document provides a list and an overview of the documents included in the PlayStation®Vita Programmer Tool Runtime Library Release 3.50 (the SDK package).

Folder Configuration of Documents

The documents in the SDK package are organized into the following directories according to language and file format.

English Version:

```
%SCE_PSP2_SDK_DIR%\documentation\en\
 pdf\SDK_doc\ (PDF format)
 chm\ (CHM format)
 Microsoft_Help_System\SDK_doc\ (Microsoft Help System format)
```

Japanese Version:

```
%SCE_PSP2_SDK_DIR%\documentation\ja\
 pdf\SDK_doc\ (PDF format)
 chm\ (CHM format)
 Microsoft_Help_System\SDK_doc\ (Microsoft Help System format)
```

The above-mentioned folders are further subdivided into the following sub-folders by categories as seen from the viewpoint of software development work.

- 1st_read
- Development_Environment
- Tutorial
- System
- Network
- Input_Output_Devices
- Audio_Video
- Graphics
- Engines
- Developer_Tools

Notes on viewing these documents

- The documents in this package can be viewed with Adobe® Acrobat® 8.0 or later, or Adobe® Reader® 8.0 or later. You can download the latest version of the Adobe Reader from the website of Adobe Systems Incorporated.
- In some HTML documents, links may be missing at locations where they should have been inserted, or activating them may cause a jump to the wrong destination.
- The documents in this package have the author/copyright holder and reference destination URL, etc. are attached when reference materials located outside of the package are shown. Reference materials that do not contain this information are documents included in the SDK document package.
- "PSP2" and "VITA" does not indicate a product name. Refer to the "Brand Guideline" for notation, abbreviations and definitions of logos in PlayStation®Vita documents.
- See the following website for trademarks:
  https://psvita.scedev.net/ext/copyright

2 Change History/Restrictions

Document Package Changes

This document provides an overview of the additions and changes that have been made in the documents of this release.

Information Resource Guide

This document introduces information that is useful for developing PlayStation®Vita applications.

Document List

This document shows a list of the documents to be referred to by persons developing software using the SDK package, and provides an overview of their respective contents.

System Software Overview

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

The sequence of contents covering the above-mentioned development support features 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.

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

Patch Overview

This document describes online patching, which updates the application for feature extensions over the network.

Patching mechanisms and the behavior of systems involved in patching are described. Also, important notes regarding the creation of applications and patch package are included.

Error Overview

This document explains how to handle the error codes in PlayStation®Vita. In PlayStation®Vita, short error codes are used instead of hexadecimal error codes for error notification to the user. A tool "PSP2 Error Code Viewer" is also provided for an efficient search of short error codes displayed through Message Dialog, etc. or hexadecimal error codes returned by the libraries.

Application Development Process Overview

This document explains the application development that uses SDK. It consists of the following main contents.

- About application package configuration
- About application configuration
- How to access files included in the main program files of an application
- How to access patch

- How to access save data
- How to access additional contents
- How to use a memory card
- How to install additional contents in an application
- How to upgrade to the full version within an upgradable application
- How to access game data and save data from the development host computer
- How to load save data created with a product version application on the Development Kit and Testing Kit
- How to create and install application package files
- About the feature for importing and exporting an application game data and other data
- Troubleshooting for issues encountered during development

Content Information Specifications

This document indicates data specifications, such as, for icon images required to display applications on the system software's home screen, and for software manuals of applications that can be displayed from LiveArea™.

DevKit/TestKit Activation User's Guide

To prevent unauthorized use of a PlayStation®Vita Development Kit or Testing Kit, the Development Kit and Testing Kit are equipped with a feature to check the expire date. This document explains the necessary procedure to set the expire date and continue using the Development Kit and Testing Kit.

3 Development Environment

Hardware Overview

This document describes the features of the hardware implemented on PlayStation®Vita. Furthermore, details on the custom SoC, which implements the system core features, are explained by providing a block diagram as well as a specification overview for each feature block.

Programming Startup Guide

This document provides basic information required for an application's development environment and programming environment. Also, it contains guidelines to be observed when creating applications. It consists of the following main contents.

- Basic information of the development environment
- Basic information and notes on the programming environment
- Termination processing of the application
- Resources that can be used by the application
- Support for PlayStation®TV by the application
- Notes on application development with consideration for the retail unit PCH-2000 series

DevKit/TestKit Setup Guide

This document explains the development environment setup procedure for using the Development Kit and Testing Kit and gives an overview of the usage methods. It consists of the following main contents.

- Procedure to install the development software (compilers, libraries, sample programs, etc.) on the development host computer (Windows PC)
- Main points of procedure to connect the Development Kit and the development host computer
- How to update the Development Kit and Testing Kit
- Procedure to compile and execute sample programs
- Troubleshooting
- Tutorials for application creation/debugging/performance analysis
- How to set up the development environment for PlayStation®TV

Development Kit Neighborhood Settings Guide

This document explains how to use host tools to configure the settings of the Development Kit from a development host computer after completing Development Kit set up. Detailed descriptions are also provided for each setting item. This document focuses on the setting method using the Neighborhood for PlayStation®Vita.

4 Tutorial

Graphics Programming Tutorial

This document explains the graphics programming for PlayStation®Vita.

This document is intended for graphics programmers with experience in fixed-feature pipeline graphics architecture such as the one used for PSP™ (PlayStation®Portable) and who will soon be using graphics in PlayStation®Vita.

Precomputation Tutorial

This document explains the precomputation feature included in libgxm. By creating precomputed objects, it may be possible to boost an application's performance. In this document, the creation methods for precomputed objects and the settings/update methods for improving CPU performance are explained.

Post-processing Tutorial

This document explains many of the post-processing in the SGX graphics chip of the PlayStation®Vita. It describes the implementation of post-processing and their performance characteristics. In addition, this document assumes the reader has basic knowledge of real-time rendering.

HDR Rendering Tutorial

This document explains about the procedures required for processing HDR (High Dynamic Range) rendering effects with the PlayStation®Vita hardware. The HDR rendering and various visual effects are explained, and code implementation examples are shown.

Deferred Rendering Tutorial

This document explains the technique for implementing deferred rendering on the GPU of the PlayStation®Vita and its performance characteristics.

Soft Particles Tutorial

In the case of rendering effects such as fire, explosions, and smoke using particles, if those particles intersect with other geometry in the scene, it creates visible hard edges, thus reducing the quality of these effects. A method to address this issue is to render soft particles to soften those intersection edges. This document describes how to implement the soft particles algorithm on PlayStation®Vita.

Water Simulation Tutorial

This document explains implementation examples of water simulation using fragment shaders common in game development. Methods for porting the CPU workload to the GPU are shown for water simulation effects.

Shadow Techniques Tutorial

Shadow is an essential component to enhance the sense of realism of games. This document describes how to produce shadows using shadow mapping techniques with libgxm features. In addition to the method to implement shadows using soft shadow and cascade shadow mapping, the method for improving the quality of shadows and performance using these shadow mapping techniques are provided in this document. By referring this tutorial, the best shadow mapping settings according to the quality or performance requirement can be selected by changing the parameters and checking the effects of them.

Hidden Surface Removal for GPU: Tutorial

This document explains how to perform hidden surface removal using the ISP (Image Synthesis Processor) unit in the GPU for PlayStation®Vita to improve rendering performance. In this tutorial, various test scenes are implemented, allowing confirmation of the effects the ISP unit processing has on the rendered geometry categories.

Video Texture Tutorial

This document explains how you can integrate the video player library into a typical game rendering engine to display video content in three-dimensional scenes.

USSE Performance and Shader Optimizations: Tutorial

The GPU of PlayStation®Vita includes a shader unit (a unified shader) called the Universal Scalable Shader Engine (USSE). The USSE is flexible in regards to vertices and fragment shaders, and enables high performance.

This document explains optimization test cases for shaders in PlayStation®Vita and statistical analysis of performance in order to easily identify bottlenecks and processing limitations for processing in applications. In addition, it contains effective tips for the complicated sections of shader programming for PlayStation®Vita and for adjusting and optimizing shaders in order to improve application performance.

Multi Thread Programming Tutorial

For execution of PlayStation®Vita applications, three CPU cores can be used; therefore, executing program processing in parallel by actively using multi threads will lead to the enhancement of the application performance. This document explains the parallel programming environment available for PlayStation®Vita SDK and how to use it.

Memory Management Tutorial

This tutorial provides explains the PlayStation®Vita memory hardware and memory-related APIs. It also discusses debugging techniques and the performance characteristics according to memory type including caching modes.

Memory Management Function Replacements of the C and C++ Standard Libraries: Tutorial, Memory Management Function Replacements of the C and C++ Standard Libraries: Reference

These documents explain how to replace the memory management functions of the standard C/C++ library with a user-defined memory management functions on the PlayStation®Vita.

View Frustum Culling Tutorial

View frustum culling technique is used to reduce the GPU load by culling undrawn objects on the CPU side. This document describes how to implement the view frustum culling through the sample programs created with the view frustum culling technique.

Algorithms for Sorting on PlayStation®Vita: Tutorial

This document explains the various sort algorithms and performance implemented in PlayStation®Vita. It is intended for engineers with detailed knowledge of sort algorithms.

Audio Tutorial

This document provides an overview of 5 types of sound libraries prepared for PlayStation®Vita SDK and describes how to use NGS, one of the sound libraries, that plays a main role when creating a sound in a game application.

Touch Measure Tutorial

This document explains the sample code that evaluates the performance of touch operations in order to implement responsive touch operations.

TRC Compliant Shooting Game Creation Tutorial

This document describes a sample program that complies with TRC (Technical Requirements Checklist), which consists of technical requirements that must be complied with by applications, and notes and other points to be observed for game implementation taking into consideration TRC. The sample program complies with the TRC requirements to be of reference for application development.

Basic TRC Compliant Tutorial

This document explains sample program that includes a minimum feature set that is implemented in a TRC (Technical Requirements Checklist) compliant way, and also explains the points to note when performing implementation with the TRC in mind. This sample application supports the single-player auto-save feature, small-scope game trophies, and LiveArea™.

Physics Effects High Level API Tutorial

This document explains how to use the physics engine that performs calculations in order for the various objects in a game to behave according to the laws of physics. The Physics Effects library that is provided with the SDK actually consists of a collection of parts that bring together physics simulation related features, and it is used incorporated into the game engine. The Physics Effects High-Level APIs also provide the following extended features:

- Buffer management
- Fixed pipeline structure
- Sleep mechanism
- Area raycasting
- AABB overlapping
- Registration of non-colliding pairs
- Multi-threading
- Debug support
- Sub-step simulation
- Collision query

System Gesture Demo

This document explains a demo sample program that is for operating butterfly-shaped objects displayed on the screen using libSceSystemGesture.

Game Orientated Camera Demo

This document explains simple camera processing code that uses libcamera and explains the sample program that obtains the orientation of a PlayStation®Vita using the motion sensors. In addition, various methods for improving performance are explained.

360-Degree Video Playback Demo

This document explains the techniques used in the sample code of the "360-degree Video Playback Demo". It also explains the unwrap processing for images taken with a 360-degree camera.

Cross-Play Tutorial

This document explains how to create a simple shooting game that uses the Cross-Play feature for PlayStation®Vita and PlayStation®3.

"Cross-Play" refers to the feature that allows a PlayStation®Vita user and a PlayStation®3 user to join the same online game session to participate in player-vs-player and cooperative gameplay. Users of PlayStation®Vita and PlayStation®3 systems can share seamless game experiences that use the various services of PSN℠ with the Cross-Play sample introduced in this document.

System Software Theme Tutorial

This tutorial provides an overview of the features, tools and production cycle for System Software Themes on PlayStation®4 and PlayStation®Vita/PlayStation®TV. It also takes a look at the launch features and comparison chart with Dynamic themes for PlayStation®3.

5 System

Kernel Overview, Kernel Reference

These documents explain the features of the PlayStation®Vita kernel. For the detailed specifications of each interface, refer to "Kernel Reference" document.

Power Service Overview, Power Service Reference

These are the power service documents. The power service is a kernel mode service that provides the following power management features for the PlayStation®Vita.

- Get the remaining percentage of battery life
- Transition to suspend and resume from suspend
- Power control
- Power button control
- Switch the dynamic operating clock frequency of the PlayStation®Vita system chip

C and C++ Standard Libraries: Overview and Reference

These are the C/C++ Standard Library (libc, libm, libstdc++) documents.

(These documents are provided only in HTML format.)

Vector Math Library Overview, Vector Math Library Reference

These documents cover the Vector Math library that provides the 3D and 4D vector operations, matrix operations, and quaternion operations used in 3D graphics. This library provides also 2D vector and matrix feature that can be useful in user interface or human interface device programming. APIs for the C ++ programming language are provided, with three formats according to the data layout:

- The scalar format, which is useful for porting to the development environment
- The AoS (Array of Structures) SIMD format, which can easily and quickly be adapted to handle different situations
- The SoA (Structure of Arrays) SIMD format, which allows for maximum throughput

Geometry Library Overview, Geometry Library Reference

These are the documents for the geometry library, which provides basic geometry primitive classes, collision detection, ray tracing, and other related functions useful for geometric calculations.

libsysmodule Overview, libsysmodule Reference

These are the documents for libsysmodule, which is required for using the libraries implemented in the PRX format. They provide also a simple overview of the PRX format library system.

OpenPSID Overview, OpenPSID Reference

These documents explain OpenPSID. OpenPSID is an device-specific ID. Because various problems may arise in connection with its use, SCE's authorization is required for OpenPSID use. OpenPSID cannot be used for purposes such as strict copy protection and user management.

libscebase Overview, libscebase Reference

These documents describe libscebase that provides datatypes and macros used in the entire PlayStation®Vita SDK system.

libdeflt Overview, libdeflt Reference

These are the documents for the libdeflt library, which is used for compressing and expanding data. Compressing and storing data and then expanding it at the time of use enables the efficient use of memory, a reduction in data loading times, etc. libdeflt supports four formats for data compression and expansion, the ZLIB format, the DEFLATE compression format, and the GZIP format, which are defined in RFC 1950, RFC 1951, and RFC 1952, and PK0304 ZIP format.

libfpu Overview, libfpu Reference

These are the documents for the libfpu library which provides mathematical functions. This library supports single-precision floating-point accuracy via FPU, and has the following differences with the mathematical functions (libm) included in the standard C library.

- High-speed and compact, does not use TLS (Thread Local Storage)
- Supports only single-precision calculations
- The calculation result is not guaranteed to be strictly correct up to the LSB
- No error notification via errno is made
- The result differs for NaN and infinity
- The precision is not guaranteed for unnormalized numbers

libheap Overview, libheap Reference

These documents explain about libheap, a library for managing in smaller units the memory allocated in advance using the kernel's memory block APIs. libheap is useful when you want to perform malloc()/ free()-like memory management. It can also be used when you want to allocate multiple independent heaps. libheap can automatically extend the block using the memory block API of the kernel if the heap memory size is insufficient.

In addition the mspace function of libc manages the memory block specified when the mspace_create() function was called in advance.

libfiber Overview, libfiber Reference

These are the documents for the libfiber library, which uses the fibers provided by the PlayStation®Vita kernel. By using fibers, application programs have multiple processing flows, and can execute processing by switching them as needed.

libatomic Overview, libatomic Reference

These are the documents for the libatomic library, which provides features for updating shared variables with indivisible (atomic) operations in parallel programs. Since libatomic executes with indivisible operations, even when a single variable is operated in parallel from several programs, the intermediate states during the processing cannot be monitored from the other programs.

libult Overview, libult Reference

These are the documents for libult, which provides lightweight user level threads and lightweight sync objects.

librtc Overview, librtc Reference

These are the documents for librtc, which is used to obtain time information. It provides the current time, which can always obtain CPU internal timer managed by the kernel safely. librtc provides various time formats and conversion functions.

libsha0 Overview, libsha0 Reference

libsha0 is a library that is used to generate a digest value in the SHA-0 (Secure Hash Algorithm) format. It can be used to detect data corruption and prevent data tampering by applying Keyed-Hashing for Message Authentication (HMAC).

libsha1 Overview, libsha1 Reference

libsha1 is a library that is used to generate a digest value in the SHA-1 (Secure Hash Algorithm 1) format as defined by RFC3417 (FIPS 180-1). It can be used to detect data corruption and prevent data tampering by applying Keyed-Hashing for Message Authentication (HMAC).

libsha224 Overview, libsha224 Reference

libsha224 is a library that is used to generate a digest value in the SHA-224 (Secure Hash Algorithm 224) format as defined by RFC3874. It can be used to detect data corruption and prevent data tampering by applying Keyed-Hashing for Message Authentication (HMAC).

libsha256 Overview, libsha256 Reference

libsha256 is a library that is used to generate a digest value in the SHA-256 (Secure Hash Algorithm 256) format as defined by FIPS 180-2. It can be used to detect data corruption and prevent data tampering by applying Keyed-Hashing for Message Authentication (HMAC).

libsha384 Overview, libsha384 Reference

libsha384 is a library that is used to generate a digest value in the SHA-384 (Secure Hash Algorithm 384) format as defined by FIPS 180-2. It can be used to detect data corruption and prevent data tampering by applying Keyed-Hashing for Message Authentication (HMAC).

libsha512 Overview, libsha512 Reference

libsha512 is a library that is used to generate a digest value in the SHA-512 (Secure Hash Algorithm 512) format as defined by FIPS 180-2. It can be used to detect data corruption and prevent data tampering by applying Keyed-Hashing for Message Authentication (HMAC).

libsha512t Overview, libsha512t Reference

libsha512t is a library that is used to generate a digest value in the SHA-512/t (Secure Hash Algorithm 512) format as defined by FIPS 180-4. It can be used to detect data corruption and prevent data tampering by applying Keyed-Hashing for Message Authentication (HMAC).

libsfmt607 Overview, libsfmt607 Reference

libsfmt607 is a library for generating pseudo random numbers with SFMT607 (SIMD-oriented Fast Mersenne Twister 607).

libsfmt1279 Overview, libsfmt1279 Reference

libsfmt1279 is a library for generating pseudo random numbers with SFMT1279 (SIMD-oriented Fast Mersenne Twister 1279).

libsfmt2281 Overview, libsfmt2281 Reference

libsfmt2281 is a library for generating pseudo random numbers with SFMT2281 (SIMD-oriented Fast Mersenne Twister 2281).

libsfmt4253 Overview, libsfmt4253 Reference

libsfmt4235 is a library for generating pseudo random numbers with SFMT4253 (SIMD-oriented Fast Mersenne Twister 4253).

libsfmt11213 Overview, libsfmt11213 Reference

libsfmt11213 is a library for generating pseudo random numbers with SFMT11213 (SIMD-oriented Fast Mersenne Twister 11213).

libsfmt19937 Overview, libsfmt19937 Reference

libsfmt19937 is a library for generating pseudo random numbers with SFMT19937 (SIMD-oriented Fast Mersenne Twister 19937).

libsfmt44497 Overview, libsfmt44497 Reference

libsfmt44497 is a library for generating pseudo random numbers with SFMT44497 (SIMD-oriented Fast Mersenne Twister 44497).

libsfmt86243 Overview, libsfmt86243 Reference

libsfmt86243 is a library for generating pseudo random numbers with SFMT86243 (SIMD-oriented Fast Mersenne Twister 86243).

libsfmt132049 Overview, libsfmt132049 Reference

libsfmt132049 is a library for generating pseudo random numbers with SFMT132049 (SIMD-oriented Fast Mersenne Twister 132049).

libsfmt216091 Overview, libsfmt216091 Reference

libsfmt216091 is a library for generating pseudo random numbers with SFMT216091 (SIMD-oriented Fast Mersenne Twister 216091).

libmt19937 Overview, libmt19937 Reference

libmt19937 is a library for generating pseudo random numbers with MT19937 (Mersenne Twister 19937).

libmd5 Overview, libmd5 Reference

libmd5 is a library that is used to generate a digest value in the MD5 (The MD5 Message-Digest Algorithm) format as defined by RFC1321. It can be used to detect data corruption and prevent data tampering through the use of Keyed-Hashing for Message Authentication (HMAC).

libhmac Overview, libhmac Reference

libhmac is a library that is used to generate a digest value in the HMAC format as defined by FIPS 198.

libfios2 Overview, libfios2 Reference

The File I/O Scheduler version 2 (FIOS2) is a C library that can be used to schedule and manage input and output requests from multiple game components. The FIOS2 can be used to help make file access within a game more efficient and reliable by automatically resolving device contention and scheduling requests to optimize I/O performance. These documents explain the features, architecture, and usage procedures of FIOS2.

PSP2PSARC User's Guide

This document describes a PSP2PSARC command line tool to create PSARC archives which can be accessed from FIOS2.

When data files are archived into PSARC archive files, file sizes are compressed. Thus load times are reduced and performance can improve dramatically.

SampleUtil Library Overview, SampleUtil Library Reference

These documents describe the SampleUtil library, which is the utility library used by the sample programs.

Sample Utilities Overview, Sample Utilities Reference

Sample utilities are a set of features that encapsulate frequently used SDK features. By using the sample utilities, initialization features and frequently encapsulated features can be easily embedded in applications.

libSceSqlite Overview, libSceSqlite Reference

libSceSqlite is a library that has added PlayStation®Vita porting layers to SQLite, a Relational Database Management System (RDBMS). Applications can use libSceSqlite to enable easy use of the operation features of the SQLite database.

These documents describe the processing procedures and related information of libSceSqlite.

CES Library Overview, CES Library Reference

CES library is a library that supports handling of the Character Encoding Scheme (CES). By using the CES library, Unicode CES changes, interconversion with the CESs of each region, etc., can be done.

These documents describe the feature to convert Unicode CES and the feature to perform interconversion between Unicode and the CES of each region.

Json Library Overview, Json Library Reference

The Json library is a lightweigth library for parsing/generating JSON documents (JSON-formatted text data). By using this library, applications can obtain the information in a JSON document as tree structured data, and in contrast, data in applications can be output as JSON documents.

libxml Overview, libxml Reference

libxml was created by extracting features from the XML library used by Sony-made embedded devices including PlayStation®, and porting them to the PlayStation®Vita environment.

These documents explain the specifications and related information of libxml.

System Software Theme Overview

This document provides information required to create a "system software theme" for customizing various elements of the home screen and start screen on the PlayStation®Vita and PlayStation®TV system software. Specifically, this document covers the following.

- Theme formats
- Elements that can be customized
- Theme creation flow
- Testing operation
- Other information

LiveArea™ Specifications

This document explains LiveArea™, which is an area provided for each application in order to share the fun of gaming among users and to enable more active communication, its screen configuration, and basic information on creating LiveArea™.

libLiveArea Overview, libLiveArea Reference

libLiveArea is a library used for updating LiveArea™. These documents explain the XML update feature provided by libLiveArea and defined in LiveArea™ of the content information zone.

Cross-Platform Application Creation Guide

This document provides an overview of the "cross-platform features" realized through coordination between each platform of PlayStation®Vita, PlayStation®3, and PlayStation®4, a summary of the features and services provided by SCE for supporting the development of applications with cross platform features, and describes the requirements, notes, etc. for implementing each feature and service.

Application Manager Overview, Application Manager Reference

These documents explain the feature to obtain the event distributed by the application manager. PlayStation®Vita discloses system calls that can be called from the application side. The application manager is a kernel module managing the state of an application, as well as performing arbitration and event distribution.

Currently, the public features are as follows.

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

BGM Port Control System Call Overview, BGM Port Control System Call Reference

In the PlayStation®Vita system, an application that obtains the BGM output right outputs data written in the BGM port as BGM. These documents explain the system calls for obtaining and releasing the BGM output right required for the application to output BGM.

Music Export Library Overview, Music Export Library Reference

These documents explain the music export library, which provides a feature to export music files held by the application to an area on a memory card that is managed by the system software.

Photo Export Library Overview, Photo Export Library Reference

These documents explain the photo export library, which provides features to register the photo files of applications to the system software. The photo export library can also be used to determine whether the photo file held by the application can be viewed using the Photos application.

Screenshot Library Overview, Screenshot Library Reference

These documents contain an explanation of the screenshot library, which allows application-side control of the screenshot feature provided in the system software. The screenshot library enables/disables the screenshot feature, superimposes copyright displays onto screenshots, captures screenshots and obtains the parameters set for screenshot images.

Clipboard Library Overview, Clipboard Library Reference

These documents explain about the Clipboard library, which provides APIs for accessing the system software clipboard. It is possible to obtain/store characters strings from the text clipboard. The clipboard can be implemented in an application's text input system by using this library in conjunction with libime.

IME Overview, libime Reference, IME Dialog Reference

IME is a software component that provides text input features, and two libraries, namely Common Dialog library and APIs, are provided. These documents explain the processing procedures and the event processing for each library.

Common Dialog Overview, Common Dialog Reference

Common Dialog library is a common library that provides features such as "Message Dialog", "Save Data Dialog" and "Network Check Dialog" to dialog with end users. These documents explain the processing procedures common to all dialogs.

Save Data User's Guide

Save data includes data for indicating a game play progress, high score information, user's character edits and setting data such as a volume setting for sound effect , button preferences, and a message display speed setting. This document explains, for example, how to use save data, usage procedures of three API groups provided for using save data (Common Dialog, library, file access APIs), a basic processing procedure using Common Dialog, and data formats of save data.

Save Data Dialog Overview, Save Data Dialog Reference

These are documents for the Save Data Dialog library that supports the realization of GUI display for the application flow consisting of saving and loading of save data. These documents explain a basic procedure for prompting users to save and load the save data, and for navigating the subsequent operations. Moreover, an explanation is provided on how to use the Save Data Dialog library to display save data for PSP™

Message Dialog Overview, Message Dialog Reference

These are documents for the Message Dialog library, which is one of the features of the Common Dialog library. The Message Dialog library supports the display of information to users by applications. Applications can use the Message Dialog library to easily implement user guidance pursuant to the TRC (Technical Requirements Checklist).

Camera Import Dialog Overview, Camera Import Dialog Reference

These documents explain the Camera Import Dialog library for taking photographs from applications. The Camera Import Dialog library is one of the Common Dialog libraries; this library enables easy implementation of the process to display a screen for capturing images and actually taking photographs. The photograph data taken with this library is not registered on the system software. Use the Photo Export library for registration to the system software.

Photo Review Dialog Overview, Photo Review Dialog Reference

These documents provide an explanation of Photo Review Dialog library, which enables the application to view photo data. Photo Review Dialog library is one of the features in Common Dialog library, and provides the following features:

- Full-screen display of 1 file
- Registration of 1 file to the system software

Photo Import Dialog Overview, Photo Import Dialog Reference

These documents explain the Photo Import Dialog library, which supports the import of photo data by applications. The following features are realized by the Photo Import Dialog library.

- Feature to display photo list
- Feature to select 1 file

Video Import Dialog Overview, Video Import Dialog Reference

These documents explain the video import dialog library, which supports the import of video data by applications. The following features are realized by the Video Import Dialog library.

- Feature to display video list
- Feature to select one video file

Store Checkout Dialog Overview, Store Checkout Dialog Reference

These documents explain about the Store Checkout Dialog library that enables applications to purchase and download (including installation) the products displayed in PlayStation®Store (the Title Store) and, to install the downloaded additional contents and to upgrade applications to the full version.

The main features provided by the Store Checkout Dialog library are as follows.

- Purchase of specified products
- Download of the specified products
- Feature to install the downloaded additional contents and upgrade applications to the full version

NP Friend List Dialog Overview, NP Friend List Dialog Reference

These documents explain about the NP Friend List Dialog library, which displays the shared play history, and allows obtainment of the NP IDs of players selected by the user. The NP Friend List Dialog library offers the following features:

- Player selection through the shared play history display

In accordance with the increase in the number of friends to 2000 in the friends lists on PSN℠, the feature for displaying the friends list and selecting a player has been moved to the NP Friend List2 Dialog library.

NP Friend List2 Dialog Overview, NP Friend List2 Dialog Reference

The NP Friend List2 Dialog library displays a friends list of maximum 2000 players and obtains the NP ID of a player selected by the user.

NP Profile Dialog Overview, NP Profile Dialog Reference

These documents explain the NP Profile Dialog library that uses the Common Dialog to display PSN℠ player profiles managed by the system.

InvitationDialog Library Overview, InvitationDialog Library Reference

These are documents for the InvitationDialog library, which is one of the features of the Common Dialog library. The InvitationDialog library provides the "invitation dialog", which has features for sending/ receiving invitations to sessions.

GameCustomDataDialog Library Overview, GameCustomDataDialog Library Reference

These are documents for the GameCustomDataDialog library, which is one of the features of the Common Dialog library. The GameCustomDataDialog library provides the "game custom data dialog", which has features for sending/receiving game custom data.

Tw Dialog Overview, Tw Dialog Reference

The Tw Dialog library supports the posting of Tweets from an application to Twitter. The application can use the Tw Dialog library to easily-prompt the user to post a Tweet by specifying the initial Tweet text or an image to attach.

Cross-Controller Dialog Overview, Cross-Controller Dialog Reference

These documents explain the Cross-Controller Dialog library, which supports device coordination.

Application Utility Overview, Application Utility Reference

These are the documents for the application utility that provides following various features for game content (a subset of a file which constitutes an application, such as save data and additional contents).

- Application event related features
- Save data related features
- Additional contents related features
- Background download related features
- Content mounting related features
- System parameter obtaining features
- Application parameter obtaining features
- Safe memory related features
- PlayStation®Store related features
- Internet Browser related features
- Save data for PSP™ (PlayStation®Portable) related features

6 Network

Network Overview

This document provides an overview of the network features of PlayStation®Vita. It covers the supported devices, connection environment, and debug related information. It explains also the group of libraries for utilizing the network features.

libnet Overview, libnet Reference

These are the documents for the libnet library, which is used to install the socket layer. libnet supports the IPv4 protocol stack to which original features are installed based on NetBSD. The differences with BSD programming and the internal operation are also explained.

libnetctl Reference

This is the reference for libnetctl, which is used to obtain the connection state, the connection information, etc of Internet communication and ad hoc communication, and to connect and disconnect ad hoc communication and communication with PlayStation®3.

libhttp Overview, libhttp Reference

These are the documents for libhttp, which supports applications that use HTTP. libhttp, which runs on libnet, transmits HTTP requests to the designated URI from applications and receives the responses to these requests.

libbase64 Overview, libbase64 Reference

These documents provide an explanation of the libbase64 library, which carries out Base64 format data encoding and decoding. The procedure for encoding data in the Base64 format and the procedure for decoding data of the Base64 format are explained.

libssl Overview, libssl Reference

These documents explain libssl and the features to obtain certificate information required for libhttp to perform HTTPS communication.

librudp Overview, librudp Reference

These documents provide an explanation of the librudp library, which supports reliable data transmission on UDP (User Datagram Protocol; RFC 768). Compared to TCP over UDPP2P, supported by libnet, librudp provides a larger number of the features required by online games; moreover, the packet header is smaller, thus achieving higher transmission efficiency.

GameUpdate Library Overview, GameUpdate Library Reference

These documents provide an explanation of the GameUpdate library, which checks for patches for an applicable application. The library accesses the patch server to obtain and analyze the patch version file.

PSPNET adhoc Library Reference, PSPNET adhocctl Library Reference

These are the references for the PSPNET adhoc library, which performs connections and disconnections with the ad hoc communication of PSP™ (PlayStation®Portable).

Ad Hoc Matching Library Overview, Ad Hoc Matching Library Reference

These documents explain about the Ad hoc Matching library (adhoc_matching), which provides services in relation to the configuration of groups of participating players (matching) in ad hoc mode. The Ad hoc Matching library provides the following services.

- Notifications for players waiting to join the group and discovering their peers
- Negotiations
- Notification of join requests
- Notification of acceptance/denial of join requests
- Group management
- Confirming the existence of players who have agreed to participate
- Notifications for players who cancel their participation agreements

near System Overview

The "near" system enables the use of the various services in games, such as, the distribution of item data among users and the search of other users who are near the user.

As an overview of the "near" system, this document provides an explanation regarding the "near" game service and the "near" application. Data that can be handled by the game program via the "near" game service and points to note for using the "near" game service are also explained.

near Compliant Application Development Process Overview

The various services of the "near" system are realized by the "near" utility and "near" Dialog utility libraries. This document gives an overview of the development process of applications using the "near" system.

near Utility Overview, near Utility Reference

These documents describe the "near" utility that enables the transfer of various types of data between the "near" application and a game program.

near Dialog Utility Overview, near Dialog Utility Reference

These document describes the "near" Dialog utility, which enables the transfer of various kinds of data between game programs and the "near" application and makes it possible to use some of the features of "near" application with the dialog format.

Teleport Library Overview, Teleport Library Reference

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

PSN℠ Overview

This is the first document that should be read by developers of titles supporting PSN℠. The various features provided by PSN℠ and development resources such as various types of libraries are introduced, and the development environment setup is explained. References for various identifiers used in relation to PSN℠ such as NP title IDs and SKU IDs are also included.

NP Library Overview, NP Library Reference

These are documents for the NP library, which serves as the base library for applications that use services provided by PSN℠. The NP library provides features required for using other PSN℠ related libraries.

NP BandwidthTest Library Overview, NP BandwidthTest Library Reference

These documents explain the NP Bandwidth Test library, which provides a feature for applications to measure the communication bandwidth between clients and the PSN℠ server. The NP Bandwidth Test library uses the libhttp for measuring the bandwidth. To minimize the load of the PSN℠ server, avoid repeating unnecessary bandwidth measurements.

NpWebApi Library Overview, NpWebApi Library Reference

The NpWebApi library is a library required for accessing PSN℠ Web APIs ("Web APIs" hereafter) from PlayStation®Vita. Normally, procedures such as obtaining the token and determining the base URL are required for executing Web APIs, but since many of these processes are internally performed with this library, applications can easily execute Web APIs by using this library. In addition, the NpWebApi library provides an interface for receiving Push events.

NP Basic Library Overview, NP Basic Library Reference

The NP Basic library is a library that provides features for exchanging messages on PSN℠ and features related to the shared play history. Since binary data up to a certain size can be included in messages in arbitrary formats, users can share each other's game play in detail.

Features related to the friends list and presence information that were provided by the NP Basic library have been removed. To obtain friends information or set presence, use the User Profile Web APIs.

NP Signaling Library Overview, NP Signaling Library Reference

These are documents for the NP Signaling library, which carries out P2P communication between users on PSN℠.

NP WordFilter Library Overview, NP WordFilter Library Reference

These documents explain the NP WordFilter library for checking whether a given text contains inappropriate comments. By using this library, it is not only possible to check the presence of inappropriate comments, but also to obtain character strings replacing the characters of inappropriate comments with '*'.

NP Auth Library Overview, NP Auth Library Reference

The NP Auth Library is a library for obtaining a ticket, that is associated with a service ID or an authorization code that is associated with a client ID, from the PSN℠ server.

By using tickets, applications can check if a user has already obtained an entitlement (entitlement type product) distributed from PlayStation®Store, and applications will be able to allow a user to consume a consumable type entitlement.

Authorization codes are used to assign access rights to user information maintained on PSN℠ servers for application servers. "Application servers" mentioned here refers to servers (typically game servers) set up by licensees. By transferring an authorization code to an application server, the application server will be able to access user information maintained by the PSN℠ server.

NP Lookup Library Overview, NP Lookup Library Reference

These documents mainly explain the following contents regarding the NP Lookup library for obtaining PSN℠ user information from the server.

- A feature to search for NP IDs from online IDs
- A feature to obtain profile information by specifying NP IDs
- A feature to obtain Avatar images from obtained Avatar URLs

NP Matching 2 System Overview

This document explains the features provided by the NP Matching 2 system and shows some examples of using the features in applications. Developers can refer to this document when designing an application. In addition, descriptions on the server management and the procedures for user registration, which is required for the application development, are also provided.

NP Matching 2 Library Overview, NP Matching 2 Library Reference

These documents explain the NP Matching 2 library for using the NP Matching 2 system provided by PSN℠. This library provides, for example, features for enabling messaging and the sharing information among users who are connected to the network.

NP TUS Library Overview, NP TUS Library Reference

These documents describe the NP Title User Storage library (NP TUS library), which is a library for using storage by title and by user that is available on the PSN℠ server. By using the Title User Storage service, it is possible to store 64 TUS variables for each user and TUS data for a total of 1 MiB. Also, it is possible to store data by title for up to 8 virtual users.

NP TSS Library Overview, NP TSS Library Reference

These documents describe the NP Title Small Storage library (NP TSS library), which enables use of the title small storage service provided by PSN℠. The title small storage service supports the distribution of data to be changed after title launch, such as limited events, game balance adjustments and notices. By using this service, it is possible to decrease the load on server operation (server monitoring, domain name maintenance, updates of SSL certification, etc.) compared to setting up title-exclusive servers.

Trophy System Overview

This document explains the trophy system. The trophy system is a system for recording and rewarding the completion of various missions in the game by the users. When users satisfy conditions set in advance in the game or special conditions, trophies will be awarded in accordance with these conditions. Trophies obtained by the user so far, as well as those that have yet to be obtained are displayed on the trophy collection screen. In this way, it is possible to foster the users' sense of accomplishment and motivation to continue playing.

NP Trophy Library Overview, NP Trophy Library Reference

These documents explain about the NP Trophy library. This library provides the APIs enabling applications to use the trophy system. The trophy set (trophy setting information) is installed on the PlayStation®Vita system area using the NP Trophy library. Also, it is possible to perform processing for awarding trophies to users (unlocking) in accordance with the results of game play, and to obtain trophy icon image data and acquisition information data for displaying trophy collections in the application.

NP ScoreRanking Library Overview, NP ScoreRanking Library Reference

These documents explain the NP ScoreRanking library for using the ranking server provided by PSN℠. The PSN℠ ranking server provides multiple score boards allowing the scores of individual users (players) to be recorded. With a scoreboard, it is possible to set the number of users that can be registered, the score update conditions, and set ascending or descending order for the score. Each score can be recorded with comments, and it is possible to attach replay data and character data within the allowable total capacity limits.

Activity System Overview

This document explains about the activity system, a service to share the users' play history. It is possible to post game progress and store purchase history as status on the server, and to show this status to other users. It is possible to post comments and "Thumbs up" on each status, and to perform communication among users. A status is displayed on the communication zone of LiveArea™ and the user's profile screen.

NP Activity Library Overview, NP Activity Library Reference

These documents explain about the NP Activity library, which is for posting the status of the activity service of PSN℠. The NP Activity library allows you to set status information and to post that status on the activity server. Status can also be posted in environments that are not connected to the network. When status is posted from the game while not connected to the network, it is registered and stored in the database of system software. When connection to the network is resumed, the status that has been stored will automatically be posted on the activity server by the system software.

NP Message Overview, NP Message Reference

Game boot messages are a feature allowing users who receive the message to start up an application by opening data attachments. There are 2 types of game boot message: messages to which data to invite friends and other users to the game is attached, and messages to which data created by an application is attached.

This document describes the APIs available for use as application features, and the feature for sending/receiving game boot messages using Common Dialog.

NP Party Library Overview, NP Party Library Reference

These documents provide an explanation of the NP Party library which obtains information relating to players that are using the Party application in the system software. Application can use this library to create online games that party members can join.

Facebook® Coordination System Overview

This document explains the coordination system between PSN℠ and Facebook® on PlayStation®Vita. PSN℠ provides the following features by binding Facebook accounts with Sony Entertainment Network accounts.

- Sharing trophy earning information on Facebook
- Sharing PlayStation®Store purchase information and rating information on Facebook
- Accessing user information on Facebook accounts from applications

This document further provides guidelines for creating applications using the Facebook coordination system, and explains the precautions to be taken when creating them.

NP SNS Facebook Library Overview, NP SNS Facebook Library Reference

These documents describe the NP SNS Facebook library, which supports access to APIs of Facebook® from applications. By using the NP SNS Facebook library, it is possible to obtain the authentication information (access token) for using the Facebook Graph API directly from an application, allowing the obtainment of various user information on Facebook, and operations for posting to Facebook.

NP Toolkit Library Overview, NP Toolkit Library Reference

These documents provide an explanation of the NP Toolkit library, which is a library enabling easy implementation of main PSN℠ features in applications. The main purpose of the library is to provide an easier way of creating and managing session information for the application.

PSN℠ Commerce Service Overview

This document provides an overview of the commerce service provided by PSN℠. The main purpose is to provide reference at the planning stage of titles using the commerce service. The document explains the main points on PlayStation®Store, products that can be sold/distributed, and application development.

PSN℠ Commerce Programming Guide

This document describes the development process of applications that will use the commerce service provided on the PSN℠. Representative examples are provided on how to combine the product types and attributes that can be handled by PlayStation®Store to create "products", and on what kind of processing is to be performed within an application to handle those products. Also, the document provides an overview of the development process of products distributed by PlayStation®Store and of applications implementing the feature for purchasing products from PlayStation®Store.

NP IN-GAME Commerce 2 Overview, NP IN-GAME Commerce 2 Reference

These documents explain about the NP IN-GAME Commerce 2 library. This library provides features for browsing and purchasing the products on display at PlayStation®Store (Title Store). By using the NP IN-GAME Commerce 2 library, it is possible to set a feature for accessing the Title Store in the application, and to sell data such as additional items and scenarios, as well as access rights to game servers.

7 Input/Output Devices

Controller Service Overview, Controller Service Reference

The Controller Service is a service for reading button and analog stick data from the PlayStation®Vita system and controllers, such as, the wireless controller. These documents explain how to use controllers from applications with the Controller Service.

Touch Service Overview, Touch Service Reference

These are the documents for the touch service for reading the input data from the touch panels.

These documents explain the main features, including the basic usage procedures (the cases that calls can be made properly at the VSYNC period and calls may be made at periods of two VSYNCs or more).

libcamera Overview, libcamera Reference

These are the documents for libcamera, which is used to handle the camera.

Shutter Sound Library Overview, Shutter Sound Library Reference

These documents explain the shutter sound library for playing camera shutter sounds. The shutter sound library will forcibly play shutter sounds from the internal speaker even if the user is using an external sound output device, such as a headset.

libmotion Overview, libmotion Reference

These documents explain how to use libmotion, which provides features to obtain the orientation and motion sensor values of a PlayStation®Vita, from a developer's perspective.

liblocation Overview, liblocation Reference

These documents explain liblocation that provides geographic information such as current location, using calculation devices such as GPS (A-GPS) and Wi-Fi.

8 Audio/Video

NGS Overview, NGS Reference

NGS is an audio engine designed for games. These documents explain the library usage methods for creating user-specific audio systems using the various voice definitions provided by NGS.

NGS Modules Overview, NGS Modules Reference

These documents are for using the digital signal processor (DSP) effects provided by NGS. They explain the modules that perform various DSP effect processing.

NGS Performance Measure Sample User's Guide

This document explains the sample application that measures NGS performance. This sample application is designed to be able to measure the performance of various NGS settings. In addition, the source code is provided so that users can learn detailed knowledge relating to NGS settings and make changes for more specific examples.

NGSQuickSynth Library Overview, NGSQuickSynth Library Reference

These documents provide an explanation regarding the NGSQuickSynth library. The NGSQuickSynth library provides a method to easily set and update audio, and enables the application to directly access and control NGS.

ATRAC9™ Simple Streaming Tutorial

This document introduces a simple interface to control streaming for playing, seeking, and looping. It also explains how to implement a feature for playback of ATRAC9™ audio assets.

Audio Panning Tutorial

This document provides an overview of multiple audio panning solutions, ranging from a basic implementation of a 2D panning algorithm to more complex 3D implementations.

Streaming Controller Tutorial

This document shows how to implement streaming control for any of the following formats: ATRAC9™, VAG or PCM.

Scream Library Overview, Scream Library Reference

Scream is a runtime library for delivering interactive audio content for the PlayStation®Vita and PlayStation®4 platforms. It controls the underlying audio synthesizer, and coordinates playback of Sounds contained in Banks (the basic data format used by Scream).

Sndstream Library Overview, Sndstream Library Reference

Sndstream is a library and runtime component for playing audio files on the PlayStation®Vita and PlayStation®4 platforms. Sndstream manages tasks such as reading audio files from recorded media, decoding compressed audio data, and filling audio buffers for playback. Sndstream operates as a plug-in to Scream. Sndstream requires Scream and the NGS synthesizers for audio playback.

Screamserver Library Reference

This document provides reference information on Screamserver library. By creating a special test build of an application that includes the Screamserver library, audio designers are enabled to perform in-game editing while being live-connected to the application from Scream Tool. This allows audio designers to interactively adjust Sound properties, test Sound scripts with running an application, and mix game audio elements.

Scream Audio Glossary

This document provides a reference to terms and file extensions used in the Scream Tool and Scream Runtime documentation.

Scream BankMerge/Build Utility User's Guide

The BankMerge utility and BankBuild utility are command line tools for merging and exporting banks, and for converting to a binary runtime file. This document provides instructions on how to make settings to use these tools, as well as instructions on how to use them.

libsulpha Overview, libsulpha Reference

libsulpha is a library for measuring and analyzing audio information. Such information can be captured from NGS and then analyzed using the Windows Sulpha Tool. The Sulpha system allows users to trace problems or artifacts by giving them the ability to analyze the captured audio library function calls (along with their parameters) and to hear (and see) the audio data that was playing at that time. This provides a fast and precise method of debugging audio problems.

Audio Input Function Overview, Audio Input Function Reference

These are the documents for the library for capturing PCM format audio data. It describes how to capture the signals of audio input devices (internal microphone, headset microphone, etc.).

Audio Output Function Overview, Audio Output Function Reference

These are the documents for the low-level audio library for outputting audio in the 16-bit linear PCM, 2-channel stereo, and 1-channel monaural format. They explain the basic procedure for performing audio output and usage precautions.

libvoice Overview, libvoice Reference

These documents cover libvoice, which provides the APIs for using audio services such as CELP codec audio data compression/decompression, the optimization of mixing and volume, and audio I/O to/from headphones and microphones.

libvoiceQoS Overview, libvoiceQoS Reference

libvoiceQoS provides a C function API for implementing voice chat with Quality of Service (QoS) feature to ensure the priority and integrity of voice data. libvoiceQoS handles voice quality on behalf of libvoice, which is a voice data-handling library without QoS control. With libvoiceQoS, the voice quality is automatically managed at best effort. In addition, libvoiceQoS offers simpler APIs to use for VOIP development as compared to libvoice.

SAS Overview, SAS Reference

These are the documents for SAS, which performs synthesizer processing by software. These documents explain how to generate audio data with waveform data sound generation, pitch conversion, envelope processing, digital effects, or other forms of processing, through the use of waveform data prepared in the ADPCM format as phoneme data.

libsndp Overview, libsndp Reference

libsndp is a service for using sound data prepared with the sound artist tool. These documents give an overview of the voice management feature and the sequencer feature of libsndp. For information on the sound data (PHD/PBD) used by libsndp, refer to the "PHD/PBD Formats" document.

libcodecengine Overview, libcodecengine Reference

These documents describe libcodecengine, which provides operations relating to Codec Engine, the media processor for PlayStation®Vita. The APIs provided by libcodecengine can be roughly divided into APIs to allocate/free memory for the various libraries using Codec Engine, which can run on retail units, Development Kits or Testing Kits, and APIs measuring Codec Engine's processor load, which can run on Development Kits or Testing Kits only.

PHD/PBD Formats

This document explains about the PHD and PBD sound data formats for PlayStation®Vita. PHD and PBD are sound data formats created with the sound artist tool. This document explains the entire structure of the data format and its attributes. It also covers the PEF data format of the effect parameter.

libaudiodec Overview, libaudiodec Reference

These are the documents for libaudiodec, which decodes various types of audio data into the PCM format.

These documents explain about decoding of the following audio codecs.

- ATRAC9™
- MP3
- AAC
- CELP

libaudioenc Overview, libaudioenc Reference

These documents describe libaudioenc, which encodes PCM audio data. These describe the basic procedure for encoding and the restrictions regarding the buffer area used during encoding.

PlayStation®Vita Movie Data Creating Guidelines

This document describes the procedure for creating PlayStation®Vita movie data using a video production application with Vegas Pro as an example.

Video Player Library Overview, Video Player Library Reference

These are the documents for the video player library, which is used for playing back MP4 files containing AVC video and AAC audio streams. These documents provide the basic procedures of the library and the restrictions.

Mp4Rec Library Overview, Mp4Rec Library Reference

These documents are for the Mp4Rec library, which records video and audio input to MP4 format video files. The basic procedures and restrictions are explained.

libatrac Overview, libatrac Reference

These documents explain about the libatrac library, which is a streaming library for ATRAC™. By using libatrac, streaming playback of ATRAC9™ data and of loop sound sources can be easily implemented in an application. In addition, libatrac provides features such as changing the playback position of ATRAC9™ data, and playing-back data with an epilogue part after ATRAC9™ data loops.

AVC Decoder Overview, AVC Decoder Reference

These documents describe the AVC Decoder library, which decodes AVC Elementary Stream data. These also describe the basic procedures and restrictions of the buffer region used for decoding.

JPEG Decoder Overview, JPEG Decoder Reference

These documents cover the JPEG decoder library (scejpegdec), which provides features for decoding JPEG data. The JPEG Decoder library also provides the color space conversion feature from YCbCr444/422/420 format to RGBA format. These documents explain basic decoding procedures, procedures that depend on color space conversion during decoding, and restrictions and procedures for decoding non-YCbCr444/422/420 format images as well as images outside the range of 64 x 64 pixels to 2032 x 1088 pixels.

JPEG Encoder Overview, JPEG Encoder Reference

These documents describe the JPEG Encoder library (scejpegenc), which provides features for JPEG data encoding. These describe the basic procedure for encoding image data in YCbCr422/420 format and the buffer restrictions for positioning I/O data that results from processing on a Codec Engine.

SCEPNG Overview, SCEPNG Reference

These are the documents for the SCEPNG library for handling image files in the PNG format (defined by RFC-2093). These documents describe basic encoding and decoding procedures, and encode/decode functions.

9 Graphics

Display Service Overview, Display Service Reference

These documents describe display service that provides features such as setting of the address of the frame buffer to be output to the display and flip control of the frame buffer.

GPU User's Guide

This document provides an overview of the SGX543MP4+ which is used as the GPU of a PlayStation®Vita and explains the supported data formats. The document provides a high-level overview of GPU processing and data flow and describes each hardware block's primary features.

In addition, performance data and basic checkpoints are explained for efficient GPU programming.

libgxm Overview, libgxm Reference

These documents explain about the low-level graphics library libgxm for PlayStation®Vita. They provide an overview of the SGX architecture used for creating graphics and explain the usage procedures of the main APIs (rendering API, shader API, shader patcher API).

Shader Compiler User's Guide

This document contains information related to a shader compiler that supports Cg 2.2 and the profiles supported by the shader compiler.

psp2shaderperf User's Guide

This document explains the usage of psp2shaderperf which is provided as a command line utility for parsing shaders.

Texture Pipeline User's Guide

The GXT format is a file format used for saving texture data used for libgxm. This document explains the GXT conversion tool for converting DDS/PVR/TGA format files to the GXT format and explains the PVRT compression tool for compressing DDS/TGA format files into the PVR format.

Texture Tools Reference

This is a reference for the texture tools provided as DLL (Dynamic Link Library) versions of the GXT conversion tool and PVRT compression tool, which are explained in the "Texture Pipeline User's Guide" document.

libgxt Overview, libgxt Reference

These are the documents for libgxt, which is used to perform operations when executing GXT files created using the GXT conversion tool. In addition to explaining the GXT file format, these documents also cover the runtime libraries that can be used when extracting information from GXT files.

libpgf Overview, libpgf Reference

These are the documents for libpgf, which is for using glyph images of PSP™-compatible grayscale dot font from applications. These documents explain the placement of proportional fonts using libpgf.

libpvf Overview, libpvf Reference

These documents are for libpvf, which provides glyph images of vector fonts that are built onto the PlayStation®Vita to application programs. By using libpvf, application programs can obtain the data of the vector fonts implemented in the PlayStation®Vita system.

libdbgfont Overview, libdbgfont Reference

These documents explain about the libdbgfont library, which is used for displaying the character strings on a PlayStation®Vita screen for debugging.

colladaRenderUtil Overview, colladaRenderUtil Reference

These documents describe the colladaRenderUtil library that supports the rendering of scenes defined within COLLADA.

10 Engines

libsystemgesture Overview, libsystemgesture Reference

These documents are for the libsystemgesture library that provides the gestures used by the system software GUI. They introduce the procedure for using the five touch gestures currently supported by this library, namely Tap, Drag, Tap and Hold, Pinch Out/In, and Rotation within game applications.

libhandwriting Overview, libhandwriting Reference

libhandwriting is a library that provides handwriting recognition features and returns the recognition result based on the handwriting input information. These documents explain the processing procedure of libhandwriting, the dictionaries that can be used, and the characters that can be recognized.

libsmart Overview, libsmart Reference

These documents explain libsmart that supports the creation of applications using AR (Augmented Reality).

libsmart is composed of the TargetTracking library and the SceneMapping library.

The TargetTracking library is a library that can recognize arbitrary images of planar objects (book covers, tarot cards, etc.) and AR Play Cards, then estimate the pose (position and orientation) of the camera in relation to the recognized objects. By using the TargetTracking library, it is possible to easily implement AR such as overlaying character graphics on top of a planar object shown in a camera image.

The SceneMapping library is a library that can obtain changes in camera images and motion sensor measurement information by moving the camera in a space, then use the results to estimate a 3D structure (scene map) of an unknown surrounding environment and estimate the pose of the camera in this 3D structure. It realizes a technology called Simultaneous Localization and Mapping (SLAM). By using the SceneMapping library, it is possible to recognize the 3D structure of the surrounding environment in which the camera exists, without having to recognize/track a specific recognition target, and to render a virtual character as if it really exists in the environment.

libface Overview, libface Reference

These documents explain libface that is a library for analyzing the data of a camera, etc. and recognizing the faces of people whose pictures appear in those images.

It detects multiple faces in an image, the position of the face area, its size, its angle, and its shape. It also identifies the positions and shapes of the eyes, eyebrows, nose, and mouth in each face area, the degree to which the face is smiling, the degree to which the eyes are open, the gender, whether or not glasses are worn, and an estimated value of age. Furthermore, the library compares the detected face attributes with registered data in order to identify and track individuals.

Physics Effects Overview, Physics Effects Reference

These are documents for physics simulation libraries, which reproduce a behavior of an object. These documents explain object representation methods of the Physics Effects library, guidance for implementing processing stages such as a broad phase and collision detection, and an overview of a raycast and sort processing, etc.

Physics Effects Viewer User's Guide

This document explains about the Physics Effects Viewer, a physics simulation viewer that runs on Windows. The Physics Effects Viewer loads the physics scene data of applications as snapshot files output by the Physics Effects library, and allows visual verification of behavior, physical attributes and shapes of rigid bodies on Windows. To this purpose, it can be used when desired behavior cannot be obtained on retail units, or to investigate behavior whose cause is unclear.

Physics Effects Tools User's Guide

This document explains the plug-ins used to output files from 3ds Max and Maya of Autodesk, Inc., in XML format for the use with Physics Effects.

11 Developer Tools

Core Dump Overview

This document explains the core dump feature and core file upload feature.

The core dump feature outputs information that is useful for identifying the cause when a program generates an exception. This output information includes the memory content, context, and thread state at the time of the exception. The output core file can be used by the application developers to fix bugs and to improve quality.

The core file upload feature performs a supplementary feature for the core dump feature that sends the generated core file to a specified web server.

Basic knowledge and understanding of PlayStation®Vita kernel is required in order to read this document.

Crash Reporting System Overview

This document describes the overview for the Crash Reporting System that sends crash data from a PlayStation®Vita to a server if a crash should occur due to an exception in an application that has already been released. Application developers can view the collected crash data to analyze and fix problems.

This document first describes the components and behavior of the Crash Reporting System, and the workflow for using the system. Next, it describes the necessary steps for implementing the crash detection feature in the application, and provides an overview of using the Crash Reporting Server where crash data is accumulated.

Coredump Library Overview, Coredump Library Reference

These documents explain the Coredump library. The Coredump library is a library for programs to write arbitrary information as part of the core file. In addition to the basic information included in the core file, application-specific information useful in debugging will be written, allowing confirmation of the status at the time an exception occurs.

libperf Overview, libperf Reference

These documents explain about the libperf library, which is used to analyze program performance. They explain the method to measure performance by obtaining the event counter value and cycle counter value from the ARM processor using libperf.

DTrace Overview, DTrace Reference

The DTrace tool examines the operation of the OS and applications on the Development Kit, based on the DTrace, open source developed by Sun Microsystems, Inc. These documents describe how to use DTrace, give an overview of the DTrace architecture, and provide script examples.

libdbg Overview, libdbg Reference

libdbg is a library that offers debug logging message output and assertion test features, which can be used on both the PlayStation®Vita and Windows platforms. These documents explain the libdbg usage procedures from the viewpoint of developers.

libdeci4p Overview, libdeci4p Reference

libdeci4p is a library on the Development Kit side that is used to support data exchanges with the development host computer. Target Manager APIs are provided on the development host computer side. These documents describe how to send/receive data using libdeci4p and the Target Manager APIs.

File Path Setting Guide

This document explains the file path setting file, which is used for specifying the path of the working directory during program execution and the overlay settings. By using this file path setting file, the data folder to be accessed can be switched to a different directory and only the latest updated files can be loaded from the development host computer without changing programs.