Cross-Platform Application Creation Guide

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Table of Contents

1 Cross Platform Features

This chapter contains an overview of the cross platform features and a summary of the features and services provided by SCE in order to support development of applications that have the cross platform features. Each of the following chapters explains details of each feature/service, requirements and notes on implementation, etc.

Cross Platform Feature Overview

"Cross platform features" is a term for users that refers to characteristic video game experiences realized through the linking of the PlayStation®4, PlayStation®Vita, and PlayStation®3 platforms. Specifically, the following four video game experiences are collectively called cross platform features.

- Cross-Play: PlayStation®4 users, PlayStation®Vita users, and PlayStation®3 users participate in the same online game session and perform versus gameplay or cooperative gameplay
- Cross-Controller: Using the touchscreen or camera of a PlayStation®Vita to control a large screen/high sound quality game on a PlayStation®4/PlayStation®3 or other similar situation. Through linked operation of the respective platform's applications, it is possible to provide various video game experiences that utilize two-way features.
- Cross-Save: Continuing progress of a game played on different platforms by using save data saved in online storage
- Remote play: Using a PlayStation®Vita to operate an application running on a PlayStation®4/PlayStation®3 from a distance

Note
This document refers to PlayStation®4/PlayStation®Vita/PlayStation®3 as "each platform" when there is no particular need to distinguish between them. In addition, this document refers to the Developer Network websites for various platforms as "the Developer Network website" when there is no need to distinguish between them.

Features That Support the Implementation of Cross Platform Features

Cross-Platform Matching

In order to implement Cross-Play, a matching system that an application on each platform can commonly participate in is required. The NP Matching 2 service provided by PSN℠ supports matching across platforms. For details, refer to Chapter 2 "Cross-Platform Matching".

Cross-Platform Data Transmission

In order to implement Cross-Play and Cross-Controller, data transmissions between an application on each platform are required. Libraries that support intercommunication between platforms are provided for each platform. Moreover, libraries that perform P2P communication with UDPP2P and TCP over UDPP2P by establishing a signaling connection between platforms. For details, refer to Chapter 3 "Cross-Platform Data Transmission".

Cross-Platform Online Storage

In order to implement Cross-Save, online storage that can be commonly accessed by an application on each platform is required. In addition, such online storage is also useful for implementing Cross-Play.

The PSN℠ title user storage service provides a feature for accessing online storage per user that is common between platforms. Moreover, the PSN℠ title small storage service provides a feature for accessing online storage per title that is common between platforms. For details, refer to Chapter 4 "Cross-Platform Online Storage".

Cross-Platform Ranking

A scoreboard that can be commonly accessed by an application on each platform is useful for implementing Cross-Play. The PSN℠ ranking service provides a feature for accessing a common scoreboard between platforms. For details, refer to Chapter 5 "Cross-Platform Ranking".

Cross-Platform Trophy System

In order to implement Cross-Save, sharing information related to trophies, in other words information on the trophy set and trophies that have been earned by users, between an application on each platform is required. In the PSN℠ trophy system, if the same trophy set is used across platforms, information on trophies earned by users will be appropriately merged. For details, refer to Chapter 6 "Cross-Platform Trophy System".

Note
There are still not many examples of cross-platform applications, so this document does not sufficiently provide helpful information and precautions for game design and the development/testing processes. If you have any questions, contact SCE via the Developer Network website.

2 Cross-Platform Matching

The NP matching 2 service provided by PSN℠ allows an application on each platform to share the same server and perform matching across platforms. This chapter explains procedures for settings for cross-platform usage of the NP matching 2 service with server management tools, and notes on implementation.

Settings With the Server Management Tools

NP matching 2 server operations such as lobby settings and creation can be made with the Matching2 Tool, which is one of the server management tools (PSN℠ Server Management Tools: abbreviated as "SMT"). This tool is used commonly regardless of the platform.

On the other hand, the Matching2 Service Management Tool that sets the operation status of the NP matching 2 service is provided separately for each platform. For example, in order to check the behavior of a title when access is attempted during server maintenance, perform the settings and test the operation with the Matching2 Service Management Tool for each platform.

Notes on Implementation

There are not any particular differences in programming for cross-platform matching and matching for a single platform.

To display platforms of each lobby/room member when matching, the platforms can be obtained from their NP IDs by using `sceNpGetPlatformType()`/`sceNpUtilGetPlatformType()`.

3 Cross-Platform Data Transmission

The following three methods are provided in order to perform data transmission across platforms.

- InGame messages
- P2P communication using a connection through NP matching 2
- P2P communication using a connection through signaling

Cross-Platform Data Transmission APIs

By using the following libraries, the PlayStation®4, PlayStation®Vita, and PlayStation®3 can perform data transmission across platforms.

Method		PlayStation®4	PlayStation®Vita	PlayStation®3
InGame messages		NpInGameMessage library (InGame messages)	NP Basic library (InGame data messages)	NP basic utility (data messages)
P2P communication using a connection	Through NP matching 2	NpMatching2 library	NP Matching2 library	NP matching 2 utility
	Through signaling	NpSignaling library	NP Signaling library	NP signaling utility

Specification and Obtainment of Platform

When performing cross-platform data transmission, specification of the platforms is required in addition to the specification of online IDs upon specification and obtainment of the communication peers. Specification of the recipient platform and obtainment of the platform of the sender are done using the following APIs.

PlayStation®4: Np library

- Specify the recipient platform: `sceNpSetPlatformType()`
- Get the platform of the sender: `sceNpGetPlatformType()`

PlayStation®Vita: NP library

- Specify the recipient platform: `sceNpSetPlatformType()`
- Get the platform of the sender: `sceNpGetPlatformType()`

PlayStation®3: NP library

- Specify the recipient platform: `sceNpUtilSetPlatformType()`
- Get the platform of the sender: `sceNpUtilGetPlatformType()`

InGame Messages

The following APIs support InGame message sending/receiving across platforms.

PlayStation®4: NpInGameMessage library

- Send an InGame message: `sceNpInGameMessageSendData()`
- Receive an InGame message: `SceNpInGameMessageEventCallback`

PlayStation®Vita: NP Basic library

- Send an InGame data message: `sceNpBasicSendInGameDataMessage()`
- Receive an InGame data message: `SceNpBasicInGameDataMessageEventHandler`

PlayStation®3: NP basic utility

- Send a data message: `sceNpBasicSendMessage()`
- Receive a data message: `SCE_NP_BASIC_EVENT_MESSAGE`

> **Note**
> When using `sceNpBasicGetEvent()` in PlayStation®3 systems to receive `SCE_NP_BASIC_EVENT_MESSAGE` for InGame messages to PlayStation®3 systems from PlayStation®4 or PlayStation®Vita systems, the `SceNpOnlineName` *`name`* and `SceNpAvatarUrl` *`icon`* members in the 2nd argument `SceNpUserInfo *`*`from`* will not have valid values.

P2P Communication Feature Using a Connection

The P2P communication feature with the NP matching 2 service and the P2P communication feature with signaling have a connection establishment feature for configuring P2P communication, which can be used to establish connections across platforms. After establishing the connection, obtain the IP address and port number of the peer and perform communication using UDPP2P and TCP over UDPP2P.

Reference Materials

In addition to the previously mentioned library documents, refer to the following documents for UDPP2P and TCP over UDPP2P.

- PlayStation®4: Network Overview
- PlayStation®Vita: Network Overview
- PlayStation®3: (shown in the NP signaling utility documents)

Procedure for Sending Data

(1) Obtain the companion platform

Unlike titles limited to one platform, cross-platform titles must be aware of the companion platform when performing processing. Once the companion NP ID has been obtained, use `sceNpGetPlatformType()`/`sceNpUtilGetPlatformType()` to identify the platform.

> **Note**
> With some libraries, an NP ID that does not include platform information may be returned.

(2) Specify the recipient platform

When sending InGame messages, or when using signaling to establish a connection, specify the recipient platform to the NP ID of the peer by using `sceNpSetPlatformType()`/`sceNpUtilSetPlatformType()`.

The following is an example of sending from a PlayStation®4 to a PlayStation®Vita.

```
extern SceNpId npId;

int ret = sceNpSetPlatformType(&npId, SCE_NP_PLATFORM_TYPE_VITA);
if (ret < 0) {
    // Error handling
}
// Message sending processing
```

If this specification is not performed, there is a danger of sending a message to an unintended recipient. In order to prevent such mistakes, explicitly specify the recipient platform for cross-platform titles.

(3) Send data

Refer to the respective documents of each platform for the sending procedure for InGame messages and the sending procedure by UDPP2P or TCP over UDPP2P.

Procedure for Receiving Data

(1) Receive data

Refer to the respective documents of each platform for the receiving procedure for InGame messages and the receiving procedure by UDPP2P or TCP over UDPP2P.

(2) Obtain the platform of the sender

To check the platform of received data, use `sceNpGetPlatformType()`/`sceNpUtilGetPlatformType()`. The platform type can be obtained from the sender's NP ID.

The following is an example of checking the sender platform on a PlayStation®4.

```
extern SceNpId npId; // Sender's NP ID

int ret = sceNpGetPlatformType(&npId);
if (ret < 0) {
    // Error handling
}
SceNpPlatformType type = ret;

// Afterward use "type" to display the platform type on the screen,
// specify the recipient, etc.
switch (type)
{
case SCE_NP_PLATFORM_TYPE_PS4:
    // Processing when the sender is a PlayStation(R)4
    break;
case SCE_NP_PLATFORM_TYPE_VITA:
    // Processing when the sender is a PlayStation(R)Vita
    break;
case SCE_NP_PLATFORM_TYPE_PS3:
    // Processing when the sender is a PlayStation(R)3
    break;
default:
    break;
}
```

Notes on Implementation

NP Communication ID Handling

When performing cross-platform sending/receiving of InGame messages, the sender and the recipient must have the same NP communication ID.

Meaning of an NP ID

When supporting cross-platform play, note that there may be cases where a single user is using different types of platforms simultaneously. For applications meant for a single platform, an NP ID can be handled as information representing a user, but for cross-platform cases, you must keep in mind that an NP ID includes user information and platform information.

To identify users, compare their online IDs included in NP IDs. `sceNpCmpOnlineId()`/`sceNpUtilCmpOnlineId()` is provided for this purpose.

Also, for data transmission and other processing that requires platform identification, use `sceNpGetPlatformType()`/`sceNpUtilGetPlatformType()` to explicitly obtain the platform and perform processing according to the platform. There are also cases where the platform should be considered in regards to screen design such as display methods for other users, as well as game logic.

Connection Use Limitations

The following limitations apply to the use of connections across platforms.

PlayStation®Vita

To use a connection with the NP Matching 2 library or NP Signaling library, establish the connection with only one of these two libraries. To use the NP Matching 2 library and NP Signaling library simultaneously, disable the connection establishment feature of the NP Matching 2 library.

PlayStation®3

To use a connection with the NP matching 2 utility or NP signaling utility, establish the connection with only one of these two utilities. To use the NP matching 2 utility and NP signaling utility simultaneously, disable the connection establishment feature of the NP matching 2 utility.

NP Matching 2 Utility's Connection Establishment Function Disabling Method (PlayStation®Vita/PlayStation®3)

Specify `SCE_NP_MATCHING2_SIGNALING_TYPE_NONE` for the signaling type of the signaling option parameter structure `SceNpMatching2SignalingOptParam` stored in the request option parameters structure `SceNpMatching2CreateJoinRoomRequest` specified with `sceNpMatching2CreateJoinRoom()`.

4 Cross-Platform Online Storage

The title user storage service (hereafter "TUS service") and title small storage service (hereafter TSS service) allow a title on each platform to share the same server and save/distribute data across platforms. This chapter explains settings for cross-platform usage of these services with server management tools, and notes on implementation.

Settings With the Server Management Tools

TUS server operations such as data slot/variable slot settings can be made with the Title User Storage Tool, which is one of SMT. Access rights can be set to each slot for whether or not reading/writing by other users is permitted, but once a cross-platform usage application is accepted, it will be possible to separately set access rights of the owner and other users for each platform. For example, it will be possible to set slots such as the owner being able to read/write from a PlayStation®Vita but only being able to read from a PlayStation®3 and a PlayStation®4 while other users can only read from all platforms. For details, refer to the "Server Management Tools NP Title User Storage Tools User's Guide" document.

Note that if information that affects the earning of trophies is saved to the TUS data/TUS variable, set the slot so that other users cannot write to it from either platform (this is further explained in the "Handling Information That Affects Trophies" section later in the document).

There are no cross-platform specific settings for the TSS service.

Notes on Implementation

Data Source Platform Information

Information indicating which of the platforms wrote to the TUS data/TUS variable cannot be obtained with APIs (the NP ID of the user who last updated the data/variable is not included in the platform information). If the data source platform information is required, set an application-unique value for the accessory information (`SceNpTusDataInfo`) in the TUS data.

Data Conflicts

For a single data (TUS data or TUS variable), when multiple users perform a process where the data is obtained from the server, changes are made locally, and then it is written back to the server, the risk of one user's updated data being overwritten by another user must be considered.

For example, consider TUS data used as PlayStation®3 and PlayStation®Vita common save data. A user first obtains data from the server with a PlayStation®Vita, then places the PlayStation®Vita into standby mode after playing. If the user then obtains the data from the server with a PlayStation®3 and continues the game, the save data will branch into two versions. In this case, one will be overwritten and deleted. For example, the user can save the data to the server with the PlayStation®3, but if the user then exits standby mode on the PlayStation®Vita and saves the data to the server, the data from when the game was played on the PlayStation®3 will be overwritten and deleted.

In order to deal with such problems, conflict detection is provided when uploading TUS data or writing a TUS variable. By saving the date/time of the last update notified when the data is obtained from the server and specifying this date/time to an API the next time the data is written back to the server, if the data on the server has been updated or deleted, the error `SCE_NP_COMMUNITY_SERVER_ERROR_CONDITIONS_NOT_SATISFIED` will return and the data will not be overwritten, allowing the conflict to be detected. Data conflicts can also be determined from the user of the last update, or a combination of both the date/time and the user.

Determine an appropriate procedure for when a conflict has been detected according to the specifications of the title. Possible procedures include merging the data, displaying the timestamp and/or data content and having the user select which data to save, etc.

Handling Information That Affects Trophies

Online save data, in other words saving save data to TUS data, is an effective use of the TUS service. It is also easy to share save data between platforms, but careful owner management is required for information that affects the earning of trophies. The main requirements are shown in the following, but there are also issues that depend on the title specifications, so be sure to contact SCE via the Developer Network website in advance.

Permission Settings

Set slots for online save data so that other users cannot write to them.

Specifically, use the Title User Storage Tool of SMT to set all of the following permission information for TUS slots that save save data to 0 (read/write forbidden) or 1 (read-only).

- PS4 OthersPermission
- Vita OthersPermission
- PS3 OthersPermission

Handling of Save Data Owned by Other Users

Be careful when programming so that information loaded from save data owned by other users is not uploaded to TUS data and so that information downloaded from TUS data is not saved to save data owned by other users. The precautions for each platform are as follows.

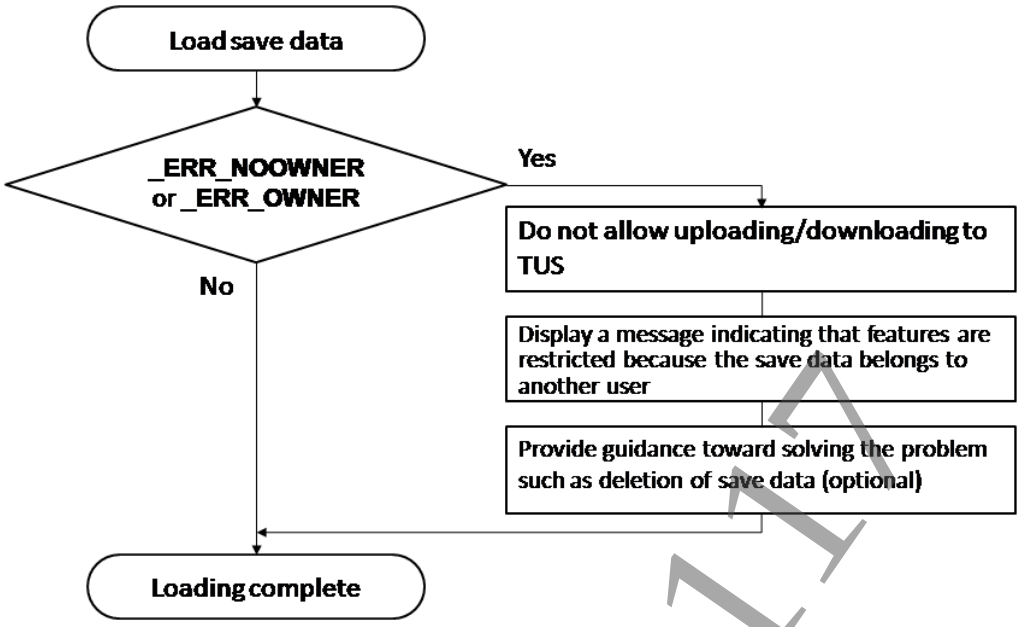
In PlayStation®3, a dedicated API for support of such restrictions is not provided, therefore applications must individually perform processing that restricts TUS access based on the owner of loaded save data.

The owner information is stored in the argument *`get->bind`* in the callback function `CellSaveDataStatGet`. When one of the following bits is stored, do not allow use of the features for uploading/downloading to TUS.

- `CELL_SAVEDATA_BINDSTAT_ERR_OWNER`
- `CELL_SAVEDATA_BINDSTAT_ERR_NOOWNER`

Accordingly, display a message indicating that features are restricted because the save data belongs to another user, and if required prompt the current user to delete the save data using the system software menu or a menu provided in the game.

Figure 1 TUS Access Restrictions Based On Save Data Ownership (PlayStation®3)

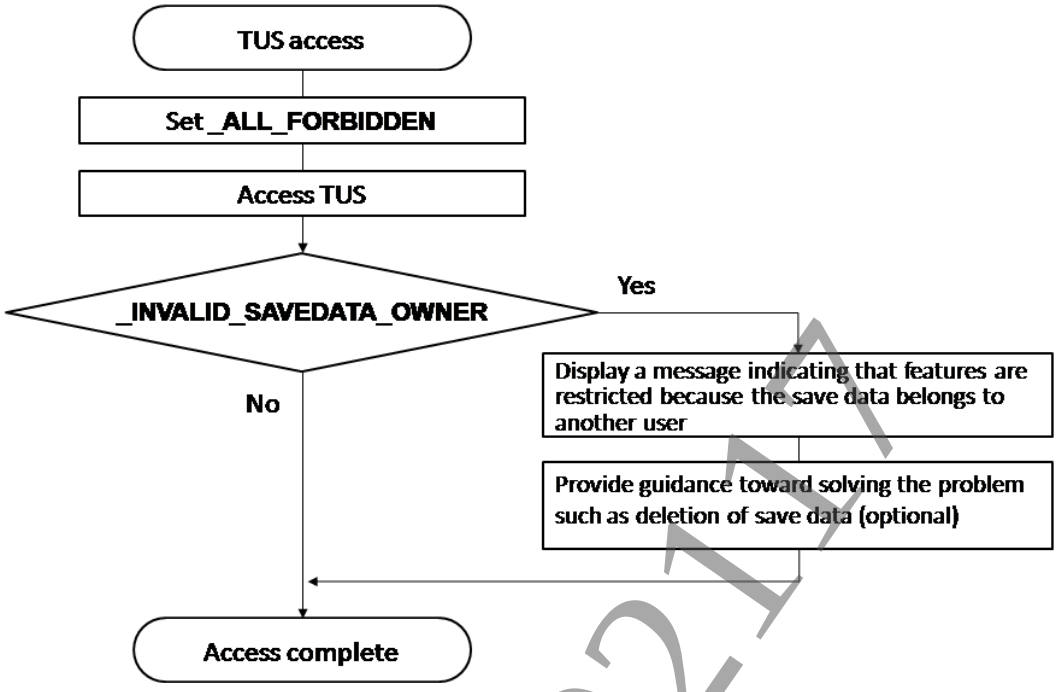


In PlayStation®Vita, in the case of a VC-VC configuration application, set restrictions according to the owner of the save data to be accessed on TUS servers (VC-MC configuration applications can also load the save data of other users, but since the MC cannot be removed, restrictions are not required). This setting can be performed by calling `sceNpTusChangeModeForOtherSaveDataOwners()` with `SCE_NP_TUS_BINDMODE_ALL_FORBIDDEN` specified for the argument *`mode`*.

Through this setting, `SCE_NP_COMMUNITY_ERROR_TUS_INVALID_SAVEDATA_OWNER` will return from communication APIs that access TUS when save data of another user is loaded, so display a message indicating that features are restricted because the save data belongs to another user. In addition, prompt the current user to delete the application in the home screen if save data resetting is desired.

Note that restricting all TUS access may cause problems from a usability perspective. Identifying information that affects trophy unlocking and restricting related requests only is preferable.

Figure 2 TUS Access Restrictions Based On Save Data Ownership (PlayStation®Vita)



In PlayStation®4, it is not possible to load the save data of another user, therefore TUS access restrictions are not required.

5 Cross-Platform Ranking

The score ranking service allows a title on each platform to share the same server with competition for scores across platforms. This chapter explains settings for cross-platform usage of the score ranking service with server management tools, and notes on implementation.

Note
Whether sharing a scoreboard across platforms is appropriate or not depends on the game. If the game has scoring that is easier depending on the platform, adding platform information to ranking displays is a good idea, but creating separate scoreboards for each platform may be better. It is also possible to create both a shared-platform scoreboard and scoreboards for each platform, then register/display all the scores. Carefully consider scoreboard usage according to each game.

Settings With the Server Management Tools

There are no cross-platform specific settings for the score ranking service.

However, when ScoreUpdateOption is set to "Update score only when breaking a personal record" with the Ranking Score Tool (Development), which is one of SMT, effects from the conflicts explained below will not be as strong.

Notes on Implementation

Registration Source Platform Information

Information indicating which of the platforms registered a score cannot be obtained with APIs. If this information is required, use the `SceNpScoreGameInfo` area and set application-unique platform information.

Score Conflicts

In the case of cross-platform scoreboard sharing, there is a possibility of conflicts occurring for scores when the same user registers scores from different platforms.

With PlayStation®Vita, it is possible to detect and prevent conflicts by using the *`compareDate`* argument of `sceNpScoreRecordScore()`, however it is also possible to create scoreboards where a problem will not be caused even if conflicts occur. For example, by setting ScoreUpdateOption to "Update score only when breaking a personal record" with the Ranking Score Tool (Development), even if scores are registered from different platforms the server will select the score with higher points, so a problem will not be caused even if a conflict occurs. It is also possible to prevent problems even if a conflict occurs by appropriately managing the save data or TUS data that is the source of the score.

The ranking servers have a delay for the score tallying, and the ranking obtained from the server is presumed to not be definite or absolute. Therefore, instead of preventing conflicts, designing conflicts to not cause problems may be an easier solution for many applications.

6 Cross-Platform Trophy System

The trophy systems for each platform support cross-platform trophy sharing. In other words, if an application on each platform has the same trophy set, when a user earns a trophy with one platform, the trophy unlock information will similar be reflected on the other platforms. This chapter explains requirements and notes for cross-platform sharing of trophies.

Games That Can Share Trophies

Cross-platform sharing of trophies is possible with games that can share a trophy set, in other words the following games.

- Games that provide video game experiences with the same content on different platforms
- Games with the same trophy earning conditions on each platform

Games that have different game content between platforms cannot share trophies.

In cross-platform compatible games where the game experience is nearly identical, the trophy sets have no changes, and the trophy unlocking conditions are unchanged, in principle the trophy sets should be shared. In particular for cross-platform compatible games that share game progress data through save data or data on servers, sharing of trophies as well is recommended.

Creation of a Trophy Pack File to Share

In order to support cross-platform trophy sharing, create a trophy pack file that is compatible with multiple platforms. When creating the trophy pack file using the Trophy Pack File Utility, there is a setting item for the compatible platforms. By selecting "`Multi (PS3™ + PS Vita)`", "`Multi (PS3™ + PS4™)`", "`Multi (PS Vita + PS4™)`", or "`Multi (PS3™ + PS Vita + PS4™)`" according to the platforms where trophy sharing is desired, you can create a cross-platform trophy pack file.

Trophy Set Updating

Uniform updates across platforms are required for cross-platform trophy sets. In other words, incrementally increase the trophy set version and select the same compatible platforms as before (or add compatible platforms), then create the trophy pack file.

It is not possible to update a trophy set for only one or some of the platforms. It is also not possible to remove compatible platforms, therefore it will not be possible to use separate trophy sets for each platform beginning with a specific version.

Example 1: Releasing a Game for Multiple Platforms at the Same Time

If a cross-platform trophy sharing game is to be released for each platform at around the same time, create a common trophy pack file, and include a copy in the master for each platform.

Example 2: Releasing an Already Released Game for a Different Platform

For example, consider a game that has already been released for the PlayStation®3 and will be released as a PlayStation®Vita version (with the same game content). If you want to share the trophies, create a trophy pack file as follows and included it in the PlayStation®Vita version master.

- Set the same trophy set and the NP Communication ID as the PlayStation®3 version
- Select "`Multi (PS3™ + PS Vita)`" for compatible platforms
- Increment the trophy set version

In this case, the PlayStation®3 version released in advance does not require additional processing. If no other modifications are required, there is no need to release a patch for only updating the trophy pack file.

If the trophy set is updated in the PlayStation®3 version in the future, note that the update must be made based on the trophy pack file included with the PlayStation®Vita version master.

Notes on Implementation

Trophy synchronization between platforms will be automatically performed by the system. Processing such as trophy set installation and trophy unlocking are the same as single-platform cases, so there are not any special considerations required from developers for cross-platform trophy sharing. Note the following points.

NP Communication ID Handling

To support cross-platform trophy sharing, the same NP Communication ID must be used between platforms.

In PlayStation®4 titles, NP Communication IDs are not directly specified in programs; by specifying an NP Service Label, the associated NP Communication ID will be used.

Save Data Handling

Be careful that the owner does not change for save data across platforms. The handling of online save data that uses the TUS service is explained in Chapter 4 "Cross-Platform Online Storage". The specifications of save data will also affect the handling, so contact SCE via the Developer Network website if you have any questions.