Patch Overview

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Table of Contents

1 PlayStation®Vita Patch System

Patch

A patch refers to the process of adding a program or data, or replacing a part of an existing program file or data file, in order to extend the feature of an application. A patch also refers to the new (or replacement) program or data used in this process.

PlayStation®Vita has a patching system for adding and replacing program files and data files of PlayStation®Vita card-released applications and applications installed to a memory card. It also allows the addition and replacement of just program files or just data files.

The main elements of the PlayStation®Vita patching system are as follows.

- Patch package
  Collects the programs and data configuring the patch.
- Patch server
  Server for registering a patch package and distributing it via the network
- Update feature
  Feature to check the patch server for a new patch package, and if found, to download and install it (In the Development Kit (DevKit) and Testing Kit (TestKit), installation is also possible locally by using the Package Installer)

Patch Package

Like an application package, a patch package is a file made with Publishing Tools collecting the programs and data for configuring a patch. There are two types of patch packages: cumulative patch packages and hybrid patch packages. These two types are explained in Chapter 2 "Creating an Application that Supports Patches". This document collectively refers to cumulative patch packages and hybrid patch packages as patch packages.

In addition, information such as the application version (APP_VER) and update information (changeinfo.xml) is included in patch packages.

Application Version (APP_VER)

Application packages and patch packages are managed by the application version (APP_VER).

The application version (APP_VER) is defined by APP_VER parameter of the PARAM.SFO file.

The version number is a 4-digit decimal number, with the 2 digits part representing the major version, and the other 2 digits part representing the minor version. A higher value indicates a newer version.

Update Information (changeinfo.xml)

Update information is used to display the details of the update when a game is updated with a patch. With update information, it is possible to check the updates contained by a patch prior to download, and to browse the details of the updates relating to all installed patches from the **Update History** menu of the home screen. Patch packages must always contain update information.

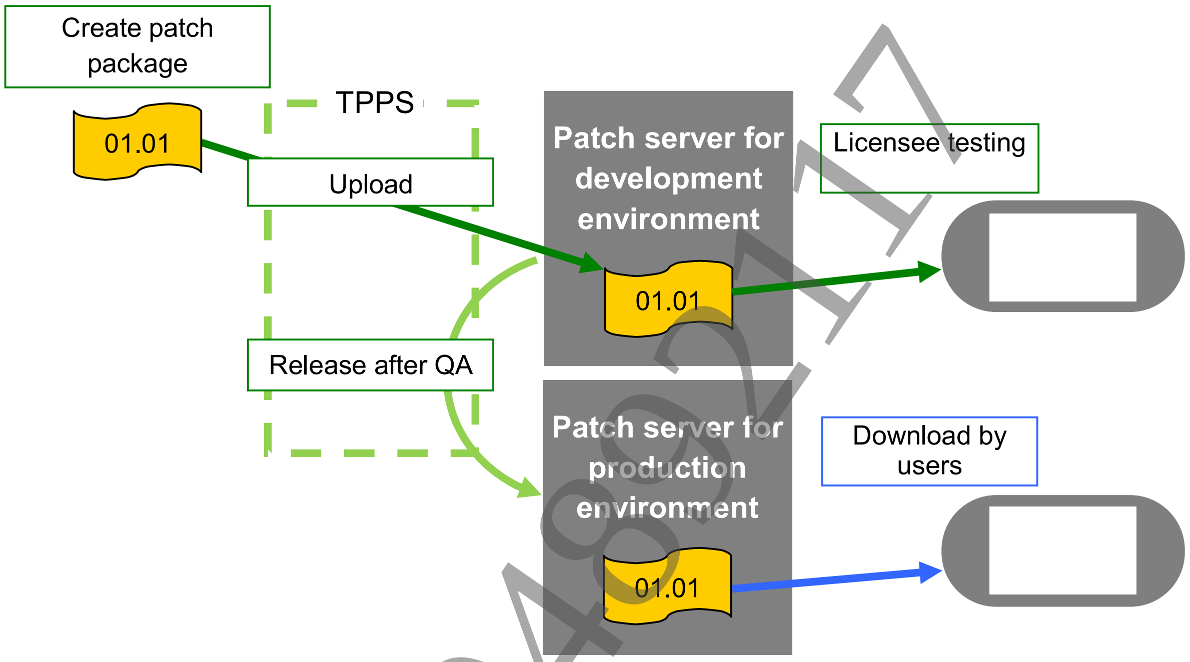
Update information is in the form of text files written in xml, and supports switching among multiple languages. For details, refer to the "Update Information" chapter of the "Content Information Specifications" document.

Patch Server

SCE provides a patch server for managing and distributing patches.

There is a patch server for the development environment, and a patch server for the production environment. Upload the created patch package, using TPPS (Title Patch Publishing System; explained later in the document), onto the patch server for the development environment. After the QA by SCE, use TPPS to deploy the patch package onto the patch server for the production environment. The patch can then be downloaded by users.

Figure 1 Patch Server



TPPS (Title Patch Publishing System)

TPPS is provided for uploading the patch package, making QA request, and also for deploying the patch package onto the patch server for the production environment. TPPS is one of the Server Management Tools (web-based application) that perform user authentication with the account of the PlayStation®Vita Developer Network website.

For details on TPPS, refer to the "Server Management Tools Title Patch Publishing System User's Guide for PlayStation®Vita" document.

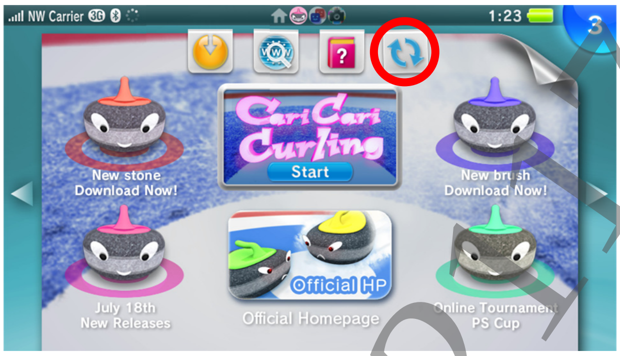
Update Feature

The update process entails checking the patch server as to whether a new patch has been registered, and if so, downloading and installing it to a PlayStation®Vita card or a memory card. The sequence of update processes, from checking to installation, is entirely performed by the system software

Checking Update

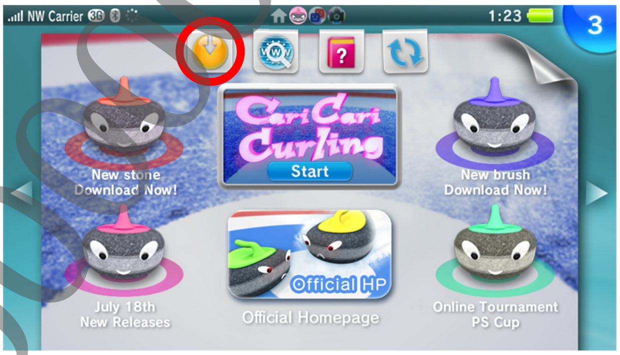
The system software checks the patch server for the existence of a patch, at the following timings.

- When the user pushes the **refresh LiveArea™ screen icon** of the content information zone of LiveArea™



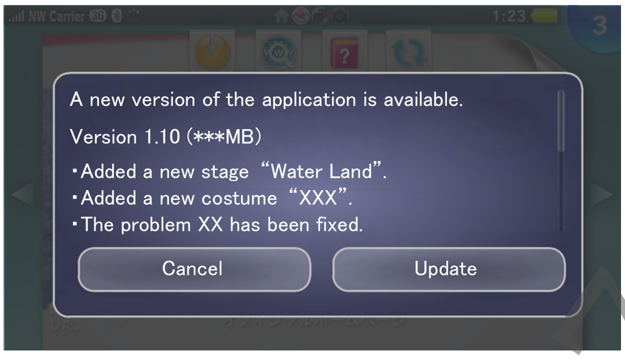
- When the application has started up Network Check Dialog in PSN℠ mode or PSN℠ online mode
- Executed in the background at a rate of once a day for the 10 most recently played titles when **Settings** -> **System** -> **Automatic Start Setting** -> **Refresh LiveArea™ Screen** is enabled.
- When the application performs patch check processing using the GameUpdate library APIs (system software 2.10 or later)
  For details, refer to the "GameUpdate Library Overview" and "GameUpdate Library Reference" documents.

If a new version patch package is available, an **Update icon** will be displayed in the content information zone of LiveArea™.



Download

Tap the **Update icon** of the content information zone of LiveArea™ to display a dialog listing the contents changed by updates. When the user selects **Update**, the download of the patch package begins.



A new version of the application is available.

Version 1.10 (***MB)

・Added a new stage “Water Land”.
・Added a new costume “XXX”.
・The problem XX has been fixed.

If a memory card is inserted, the download will be registered to the background download queue and will be processed as a background download.

If no memory card is inserted, the download will be processed as a foreground download, provided that the PlayStation®Vita card of the target application is inserted.

Installation

Downloaded patch packages are decompressed by the system software when the gate of LiveArea™ is tapped, and are installed onto the PlayStation®Vita card or onto the memory card.

If the gate is tapped while the application is suspended, a dialog will be displayed to confirm whether application is to be terminated to perform the update. If **Yes** is selected in this dialog, the installation will be executed.

When the installation is complete, the LiveArea™ screen will again be displayed.

2 Creating an Application that Supports Patches

This chapter explains the points to note when creating an application for which a patch application is planned in the future to extend its feature.

Overview

The overview of the process for creating and releasing an application that supports patches is as follows.

(1) Plan approval

Applications for which significant feature expansion through patches or release of additional contents is scheduled in advance should undergo plan approval by SCE during the planning stage, concerning the contents of the patches as well as scheduled periods and number of times.

(2) Develop and test the application

Develop and test the new version of the application.

(3) Create a patch package

When the new version of the application's programs and data is complete, identify the programs and files whose addition/update is necessary and create the patch package. Refer to the "Patches (Game Updates)" chapter of the "Application Development Process Overview" document on how to configure games and patches. Refer to the "Publishing Tools Overview" document on how to create the patch package.

(4) Perform test by installing the patch package locally

The DevKit and TestKit have a feature to install patch packages from the development host computer by using the "Package Installer". Select and install the target patch package, and check that the application is running properly after installation. For details, refer to the "Appendix: How to Install Packages" chapter in the "Application Development Process Overview" document.

(5) Perform test by using the patch server for the development environment

Use the TPPS to upload the patch package onto the patch server for the development environment. If the tests performed up to this point have yielded positive results, there should not be a problem at this stage, however, test the update and the application operation after the update.

In particular, check that all versions can be correctly updated as intended.

Note
When conducting a patch distribution test using the development environment patch server, check the configuration of the TPPS to make sure that **Server Setting (development** environment**)** is set to **ACL Setting (development** environment**)**, and set an IP address within the range of permitted addresses to the terminal running the application.
Note that if a terminal has an IP address with no access permission to the patch server, it will automatically be judged not to have any updates.

(6) Request QA

Use the TPPS to request the QA to SCE. It is possible to request QA for multiple patch packages, with different versions, at the same time.

(7) Release the patch

Once the QA approves the patch, the patch package can be moved to the patch server for the production environment. Move the patch package to the patch server for the production environment at the appropriate timing using the TPPS.

Implementation of Forced Updates

In order to forcibly ensure that the application's version is the latest when connecting to PSN℠ in applications supporting PSN℠, perform implementation as follows.

(1) Setting TPPS

In TPPS, set the **sign off flag** to **true**.

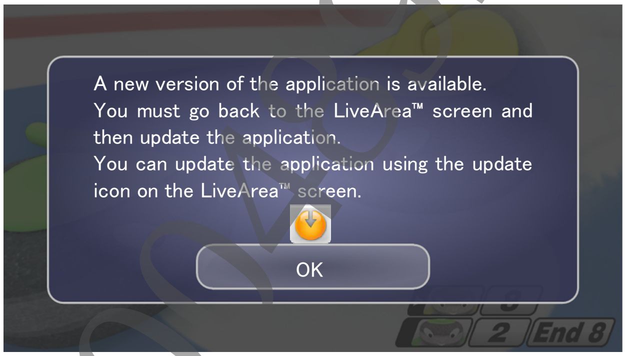
(2) Calling Network Check Dialog

At the point when you wish to check that the application's version is the latest, call Network Check Dialog in PSN℠ mode or PSN℠ online mode. Patch confirmation is performed in the processing of Network Check Dialog.

Given that, normally, Network Check Dialog is called before using the features of PSN℠, there will not be any problems provided that Network Check Dialog is implemented properly.

(3) Application Testing

If a new patch is available, Network Check Dialog will display a dialog like the one shown below.



A new version of the application is available.
You must go back to the LiveArea™ screen and then update the application.
You can update the application using the update icon on the LiveArea™ screen.

Even if a new patch is available, the Network Check Dialog processing will normally continue. However, if the **sign off flag** is set to **true**, an error will return at that moment. Applications can force the user to keep the application's version updated to the latest one by not using the features of PSN℠ if Network Check Dialog returns an error.

In the DevKit and TestKit, the above behavior can be checked without uploading the patch package on the TPPS, by enabling **Patch Test** under **Debug Settings**.

Application Version Numbering Rules

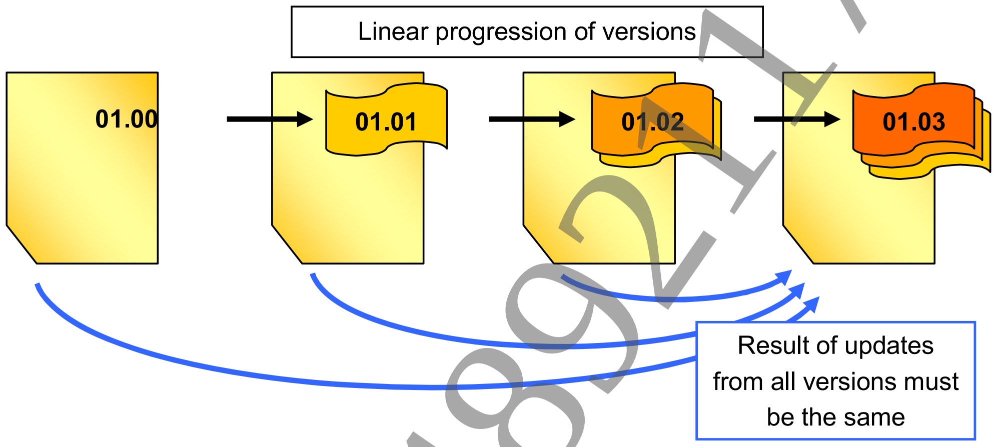
The application version (APP_VER) is in the format "*xx*.*yy*" in APP_VER of PARAM.SFO. *xx* is the major version, and *yy* is the minor version.

If the application version of the master package is "01.00", patches are numbered by incrementing the minor version by 1 ("01.01", "01.02", "01.03", ...). (Refer to the TRC [R3004])

The system software handles this value as a 4-digit number, with the period removed. The version with the larger value is judged as being newer. If APP_VER does not exist in PARAM.SFO, "00.00" will be applied as the version.

Versions progress linearly from older to newer, and there is no branching. It is necessary for all updates to a version to have the same result, regardless of the version on which the update was applied.

Figure 2 Transition of Application Versions by Applying Patches


Linear progression of versions
01.00
01.01
01.02
01.03
Result of updates from all versions must be the same

Patch Package Configuration

Cumulative patch packages and hybrid patch packages (system software 2.10 or later) are supported as patch packages.

Cumulative Patch Packages

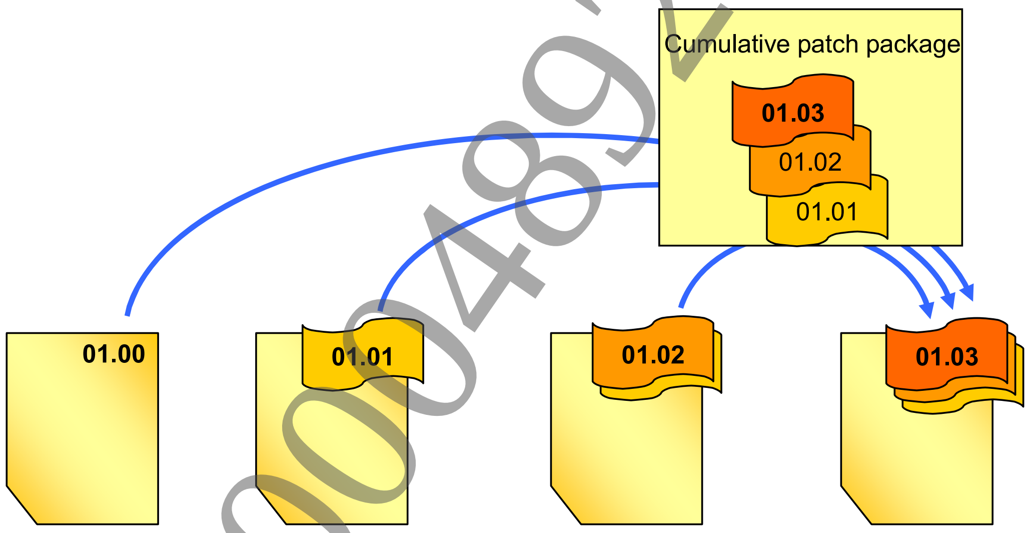
A cumulative patch package enables all versions to update to the applicable version. It includes all the additional data for the applicable version that has been released thus far.

Cumulative patch packages are patch packages that include all of the additional data from past releases required for updating to the application version of the patch itself. End users will be download all of the included additional data no matter which past application version they are updating from, but it will be possible to update to the application version of the patch itself.

When creating cumulative patch packages, all of the differential data from the first release of the application (normally application version 01.00) until the latest patch should be included in the patch package.

If the latest patch is a cumulative patch package, updates from any version released in the past are possible. Therefore, just that one cumulative patch package is placed on the patch server.

Figure 3 Application of a Cumulative Patch Package



Hybrid Patch Package

Hybrid patch packages are patch packages that place all of the cumulative data released in the past as well as the differential data for each application version on the patch server, then one is selected and downloaded. When end users perform updates, whichever of the cumulative data or differential data has a lower volume will be selected and downloaded. When differential data is downloaded, the data volume will be lower in comparison with cumulative patch packages, allowing for shorter download time.

When creating a hybrid patch package, all of the differential data from the first release of the application (normally application version 01.00) until the latest patch should be included in the hybrid patch package, similar to cumulative patch packages.

The patch server will automatically create differential data for the previous version of the patch package using the previous version of the patch package on the patch server and the newly registered hybrid patch package (cumulative data). The created differential data will be placed on the patch server as a hybrid patch package (differential data) that can be individually downloaded. When distributing the first patch, registering a cumulative patch package will always be required, therefore one cumulative patch package and one or more hybrid patch packages (cumulative data/differential data) will be placed on the patch server.

Figure 4 Hybrid Patch Package Placed on the Patch Server

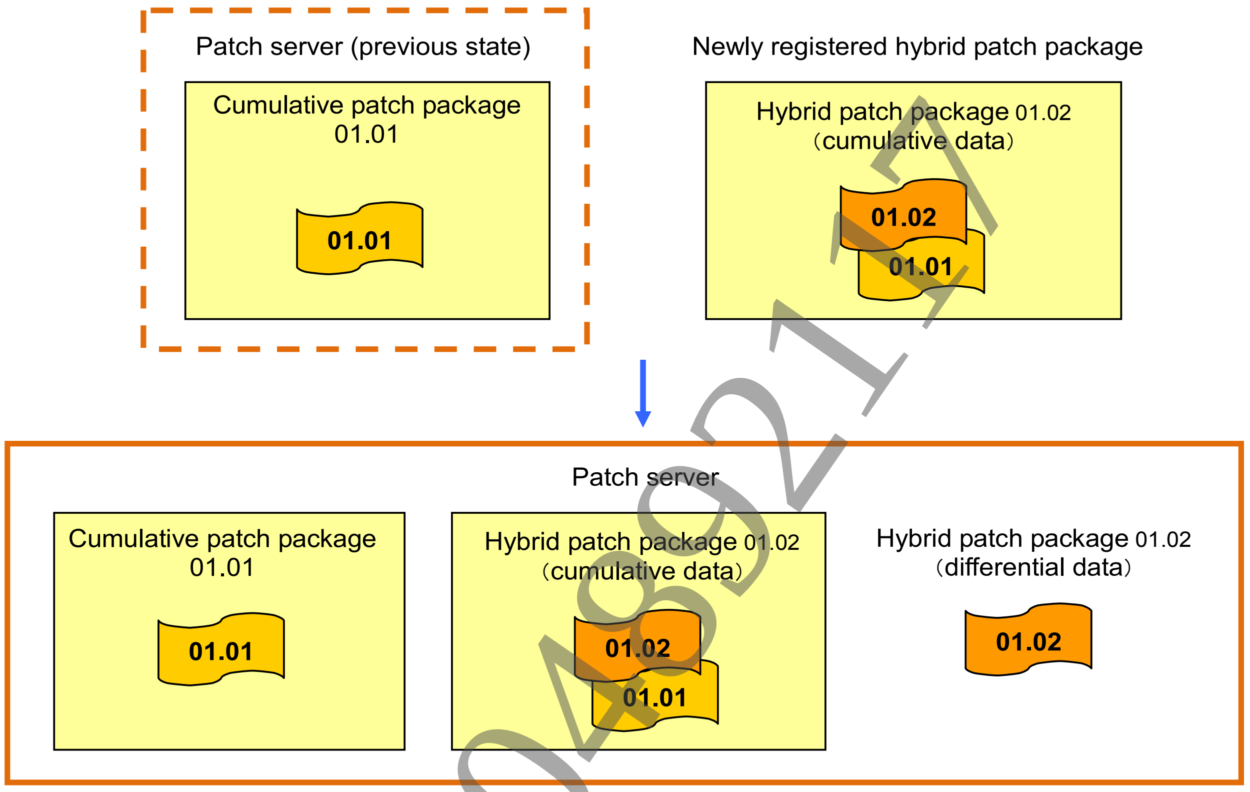

Patch server (previous state)
Cumulative patch package 01.01
01.01
Newly registered hybrid patch package
Hybrid patch package 01.02 (cumulative data)
01.02
01.01
Patch server
Cumulative patch package 01.01
01.01
Hybrid patch package 01.02 (cumulative data)
01.02
01.01
Hybrid patch package 01.02 (differential data)
01.02

Figure 4 shows the state when application version 01.02 is newly registered as a hybrid patch package on the patch server when application version 01.01 of the cumulative patch package is being distributed. The patch server automatically creates hybrid patch package 01.02 (differential data) from patch package 01.02 (cumulative data) and cumulative patch package 01.01. As a result, cumulative patch package 01.01, hybrid patch package 01.02 (cumulative data), and hybrid patch package 01.02 (differential data) will be placed on the patch server.

When a hybrid patch package is registered on the patch server, the application version (APP_VER) for the previous patch package will be automatically set based on the application version (APP_VER) set for the patch itself and displayed on the TPPS.

Figure 5 shows an example of a patch being applied when application version 01.03 of the hybrid patch package is newly distributed when cumulative patch package 01.01 and hybrid patch package 01.02 are already being distributed.

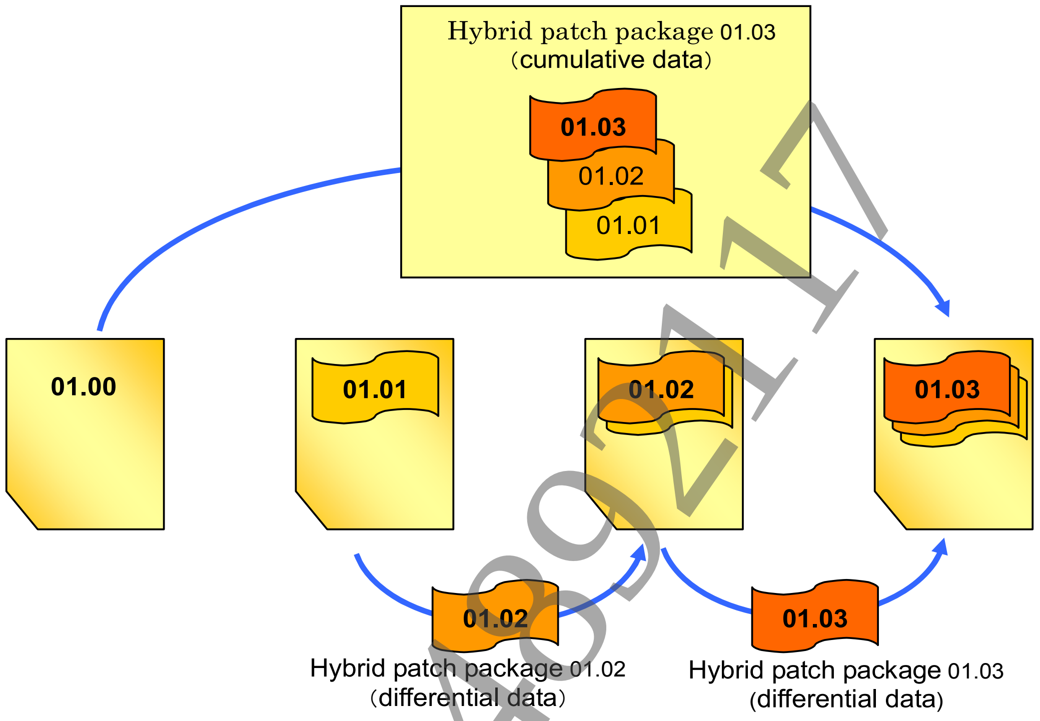
When hybrid patch package 01.03 is registered on the patch server, cumulative patch package 01.01, hybrid patch package 01.02 (cumulative data /differential data), and hybrid patch package 01.03 (cumulative data /differential data) will be placed on the patch server.

In the example in Figure 5, the patch will be applied according to the version of the application where the patch is applied as follows:

- 01.00
  Hybrid patch package 01.03 (cumulative data) will be applied individually.

- 01.01
  Hybrid patch package 01.03 (cumulative data) will be downloaded or the two hybrid patch packages (differential data) 01.02 and 01.03 will be downloaded, whichever has a lower data volume will be applied. In this example, the two hybrid patch packages (differential data) 01.02 and 01.03 will be applied.
- 01.02
  Hybrid patch package 01.03 (differential data) will be applied.

Figure 5 Example of Application of a Hybrid Patch Package



Patch Package Selection Method

At least one cumulative patch package is required for patch creation. Create a cumulative patch package when distributing the first patch. It is possible to select a cumulative patch package or hybrid patch package when creating the 2nd patch or later, but creating a hybrid patch package is recommended in consideration of patch data volumes and download times.

Patch Package Creation Method

Whether releasing a cumulative patch package or a hybrid patch package, all of the differential data up to the latest patch should be included in the created patch package. Set the type of patch package to create in Package Generator.

For details on the Package Generator, refer to the "Package Generator User's Guide" document.

3 PlayStation®Vita Card Available Space to Be Taken into Account for Patch Creation

Patches Supporting VC-VC Package Configuration

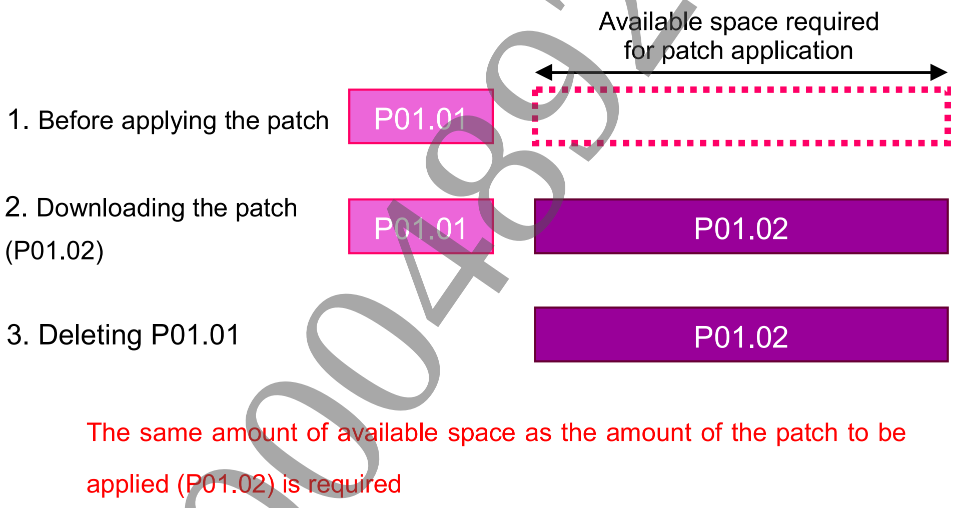
For applications with VC-VC package configuration selected in which all of an application's main data, save data, patches, and additional contents are stored on a PlayStation®Vita card, the available space in a PlayStation®Vita card is used temporarily when installing patches.

Therefore, it is required that all of the following content can be stored in the readable/writable space of a PlayStation®Vita card.

- Patches
- Temporary space for installing patches
- All additional contents
- Maximum size of save data (Save Data Quota)

The available space will be used as follows when installing patches.

Figure 6 Available Space Used for Installing Patches (Cumulative Patch and Hybrid Patch)


Available space required for patch application
1. Before applying the patch
P01.01
2. Downloading the patch (P01.02)
P01.01
P01.02
3. Deleting P01.01
P01.02
The same amount of available space as the amount of the patch to be applied (P01.02) is required

Patches that can be distributed to applications with VC-VC package configuration selected must always be installed using the space on the PlayStation®Vita card available for patch application, even when all the already distributed patches and additional contents have been installed and when the application uses the maximum size of save data (Save Data Quota) set in param.sfo.

Patch Distribution Case Study (Cumulative Patch and Hybrid Patch)

As an example, examine the following case in which patches are distributed.

Package Configuration	PlayStation®Vita Card Space	Readable/Writable (R/W) Space
VC-VC configuration	2 GB	288 MiB

It is assumed that this application uses the following space for purposes other than for patch application.

Maximum size of save data (Save Data Quota)	50 MiB
Space planned to be used for additional content distribution	38 MiB

In this case, after subtracting 88 MiB from 288 MiB, 200 MiB will be the usable size for patches and for the temporary space for installing patches.

Assumed the first patch (P01.01) distributed for this application required 30 MiB of space. Temporary space was not required for installing this first patch (P01.01).

Now consider a case where the next patch (P01.02) is distributed.

If yet another patch (P01.03) is planned for the future, the space used by the 2nd patch (P01.02) should be less than 100 MiB (less than half of the remaining space).

If the 2nd patch (P01.02) uses 90 MiB for example, installing this patch will require the total 120 MiB of P01.01 and P01.02, and installation will be possible. In such a case, the usable space for the 3rd patch (P01.03) will be 110 MiB after subtracting the space required for the 2nd patch (P01.02) from the maximum usable space of 200 MiB for patches. If future patches distributed for this application continue to use a patch size under 100 MiB, unexpected problems can also be dealt with.

On the other hand, if the space is half (100 MiB) or more than half of the empty space for patches, for instance if a 150 MiB patch is distributed, the usable space for the next patch will be 50 MiB. However, actually creating such a patch is problematic, and dealing with unexpected problems will be extremely difficult.

As explained here, it is necessary to consider future patch distribution plans for applications with VC-VC package configuration selected and then determine a patch size.

Note
Currently, submitting the master of an application with a VC-VC configuration is prohibited. Package configurations that can be used in applications are VC-MC and No VC/MC-MC (refer to TRC [R3167]).
It is, however, possible to distribute a patch corresponding to an application with a VC-VC configuration that has been released in the past. In this case as well, an application with a VC-VC configuration cannot be changed by the patch to a VC-MC or No VC/MC-MC configuration.

Patches Supporting VC-MC or No VC/MC-MC Package Configuration

For applications with VC-MC or No VC/MC-MC package configuration selected, since the temporary space required for installing patches are allocated on the memory card, the available space on a PlayStation®Vita card is not used.

4 Check for Patches by Network Check Dialog

When Network Check Dialog is called within the application, a check will be performed to see if a patch is being distributed from the patch server or not. At this time, Network Check Dialog will reference the patch check result cached from the previous patch check.

The patch check result information referenced by Network Check Dialog will be cached in the system for certain periods of time according to the following conditions.

- When a patch check is performed and a new patch is found
  Cached until the patch is installed or deleted
- When a patch check is performed and a new patch is not found
  Cached for 24 hours

The cached patch check result information will always be referred to if Network Check Dialog is called during the validity period for the cached patch check result information.

If patch check results are not cached or the cache expiration date has passed, the patch server will be queried when Network Check Dialog is called, and the cache will be updated according to the aforementioned conditions.

If always checking the state of new patches on the patch server is required, use the GameUpdate library.