SoundTrack eXporter User's Guide

For PlayStation®Vita

SDK 3.570

SIE CONFIDENTIAL

Table of Contents

©SIE Inc

About This Document

Purpose of this Document

This document describes the SoundTrack eXporter (STX) and its associated Windows tool, the SoundTrack eXporter Editor (STX Editor).

Audience

The intended audience of this document is games producers who need to know how to submit a game to QA.

Graphics

All screenshots in this guide are taken from the STX or the STX Editor tool.

Conventions

The typographical conventions used in this guide are explained in this section.

Hints

A GUI shortcut or other useful tip for gaining maximum use from the software is presented as a 'hint' surrounded by a box. For example:

Hint: This hint provides a shortcut or tip.

Notes

Additional advice or related information is presented as a 'note' surrounded by a box. For example:

Note: This note provides additional information.

Text

- Names of keyboard functions or keys are formatted in a bold serif font. For example, **Ctrl**, **Delete**, **F9**.
- File names, source code and command-line text are formatted in a fixed-width font. For example:

```
SampleProject\PSVita\images\
```

Errata

Any updates or amendments to this guide can be found in the release notes that accompany the release.

1 Introduction

This chapter introduces the SoundTrack eXporter application and the SoundTrack eXporter Editor.

SoundTrack eXporter

The SoundTrack eXporter (STX) is the application used by PlayStation®Vita users when they download game soundtracks from the PlayStation®Store. Users are prompted to install the application before it can be run. STX provides a wizard that guides the user through the exporting process.

SoundTrack eXporter Editor

The SoundTrack eXporter Editor (STX Editor) is a Windows® tool that enables games producers to gather all data needed to create STX applications and patches. Producers can reference thumbnail, splash screen and in-app images, list and edit all tracks to be included, and access the PlayStation®Vita configuration file through an external Param File Editor. When the producer has finished editing, the data is saved in a *SoundTrack project.* Producers may then create packages to be distributed on the PlayStation®Store, or patches to update existing STX applications.

The STX Editor Tool Files and Folders

Table 1 lists the folders and files that are contained in the STX Editor tool package sent to game producers.

Table 1   STX Editor Package Folders and Files

Folder	File	Description
bin\	FreeImage.dll	Tool dependency file.
	FreeImageNET.dll	Tool dependency file.
	make_full_packageVita.bat make_full_packagePS3.bat make_full_packagePS4.bat	Scripts that call Publishing Tools to build a package.
	Newtonsoft.json.dll	Tool dependency file.
	STXEditor.exe	The STX Editor Tool.
	TagLib.dll	Tool dependency file.
bin\ja\	STXEditor.resources.dll	Resource file.
bin\packageTmpVita\	dictionary.json	Localized dialog messages for the STX application.
	eboot.bin	The STX Application.
	ebootparam.ini	STX configuration file.
bin\packageTmpVita\sce_sys\	param.sfx	Default template .sfx file.
bin\packageTmpVita\sce_sys\livearea \contents\	gateIcon.png	LiveArea™ gate image.
	template.xml	LiveArea™ configuration file.
bin\packageTmpPS3\		Template directory for PlayStation®3. For more details, refer to the *SoundTrack eXporter User's Guide for PlayStation®3*.

Folder	File	Description
bin\packageTmpPS4\		Template directory for PlayStation®4. For more details, refer to the *SoundTrack eXporter User's Guide for PlayStation®4*.
docs\PSVita\	SoundTrack_eXporter-Users_Guide_e.pdf	This user guide (English).
	SoundTrack_eXporter-Users_Guide_j.pdf	This user guide (Japanese).
docs\PS3\	SoundTrack_eXporter-Users_Guide_e.pdf	The *SoundTrack eXporter User's Guide for PlayStation®3* (English).
	SoundTrack_eXporter-Users_Guide_j.pdf	The *SoundTrack eXporter User's Guide for PlayStation®3* (Japanese).
docs\PS4\	SoundTrack_eXporter-Users_Guide_e.pdf	The *SoundTrack eXporter User's Guide for PlayStation®4* (English).
	SoundTrack_eXporter-Users_Guide_j.pdf	The *SoundTrack eXporter User's Guide for PlayStation®4* (Japanese).
license\	STX_license_e.txt STX_license_j.txt	Licenses for the STX Editor Tool.
license\others\	Newtonsoft_Json_Net.txt	License for the STX Editor Tool.
SampleProject\	SampleProject.json changeinfo.xml	The STX Editor project settings. Patch update information file.
SampleProject\music	riff01.mp3-riff04.mp3	Sample music files.
SampleProject\PSVita\	param_psp2.sfo param_psp2Patch.sfo	Sample param files.
SampleProject\PSVita\images\	icon.png icon2.png inAppBackground.png liveareaBGScreen.png musicCover.jpg	Sample image files.
SampleProject\PS3\		Sample project asset files for the PlayStation®3 version. For more details, refer to the *SoundTrack eXporter User's Guide for PlayStation®3*.

SIE CONFIDENTIAL

| Folder	File	Description
`SampleProject\PS4\`	- 6 -	Sample project asset files for the PlayStation®4 version. For more details, refer to the *SoundTrack eXporter User's Guide for PlayStation®4.*

©SIE Inc

2 SoundTrack eXporter Editor

This chapter describes the main components of the SoundTrack eXporter Editor (STX Editor) UI and their functionality.

The STX Editor UI Components

Figure 1 shows the STX Editor user interface.

Figure 1   SoundTrack eXporter Editor
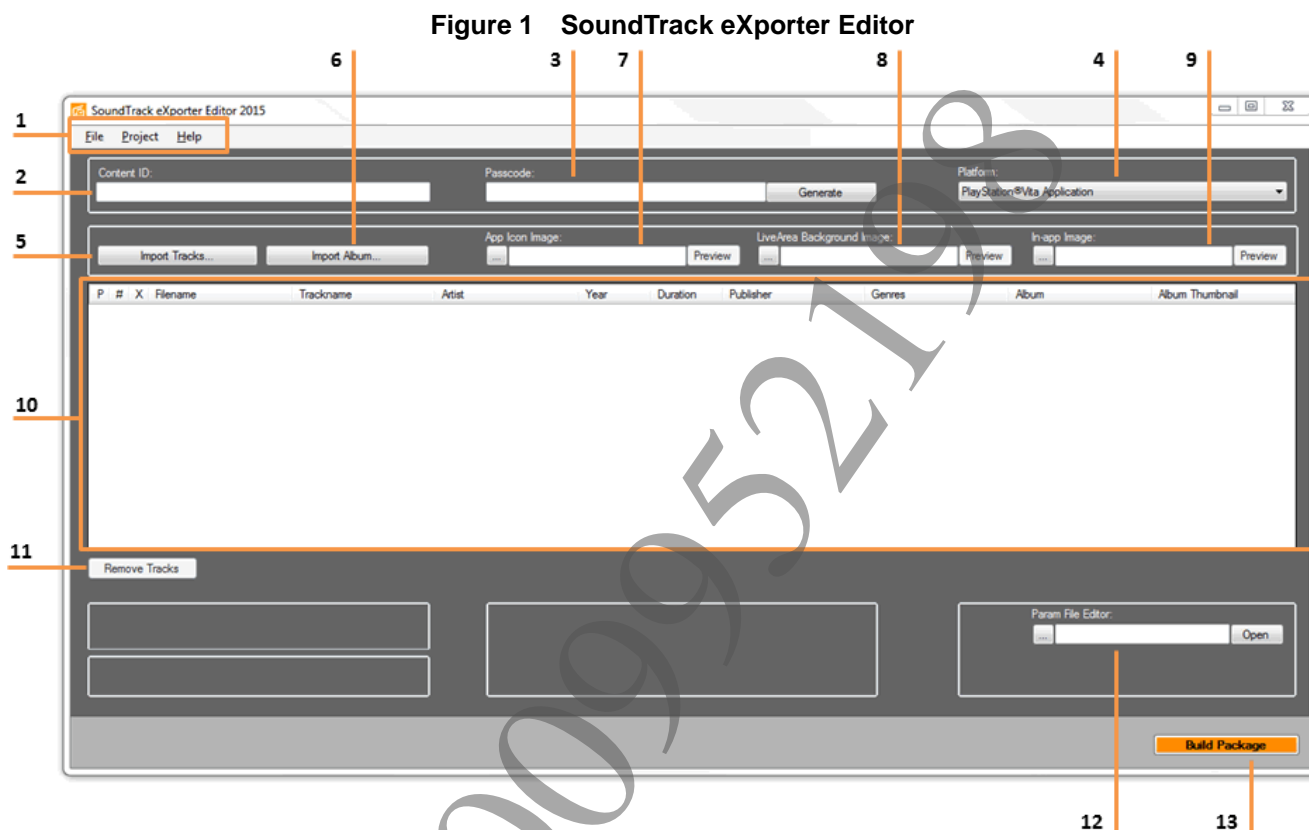
The components of the STX Editor UI are described in the following sections. The sections are numbered to correspond to the numbered highlights in Figure 1.

1. Main Menu Bar

The main menu consists of the following sub-menus:

- The **File** menu has options to **Open**, **Save**/**Save As**, **Close** and create **New** projects, and **Exit** the tool.

> **Hint:** When saving a project, you are advised to create a directory for the project to be saved in. This directory can also be used for configuration files and the final package folder, gathering all the necessary files into one place.
>
> **Note:** Avoid using space characters in file and directory paths. Some processes and applications, for example the Param File Editor, do not support directory names that contain space characters.

- The **Project** menu has options that match the tool GUI, such as options to import tracks, albums, and images, and edit the configuration file, if defined.

- The **Help** menu consists of the **About** option, which displays information about the tool, and the **Help** command, which displays this *SoundTrack eXporter User's Guide*.

2. Content ID

The Content ID is an alpha-numerical string that identifies the app once it is installed on the PlayStation®Vita. This field must match the Content ID provided to the configuration file (`param_psp2.sfo`).

For more information about Content IDs, refer to the *Publishing Tools Overview* provided with the PlayStation®Vita SDK.

3. Passcode

The passcode is an alpha-numerical string of 32 characters which is used in the package-building process.

For more information about how to set this value, see Provide the Content ID and Passcode in Chapter 4, Tutorial: How to Create an STX App Using the STX Editor.

For general information about passcodes, refer to the *Publishing Tools Overview* provided with the PlayStation®Vita SDK.

4. Platform

This dropdown list enables you to choose a platform on which to run the SoundTrack eXporter. For each platform there are two options to choose from, to build an Application or Patch build.

5. Import Tracks…

The **Import Tracks…** button launches an `.mp3` file browser from which you may choose one or multiple tracks to be imported to the SoundTrack table. If there are duplicate files in a single directory, only one instance of the file is imported.

6. Import Album…

This button launches a folder browser, from which you may choose a folder. All `.mp3` files inside the chosen folder and its subfolders are imported to the SoundTrack table. If there are duplicate files in a single directory, only one instance of the file is imported.

7. App Icon Image

This is the thumbnail image that represents the STX application on the PlayStation®Vita home page.

For more information, see Select Images in Chapter 4, Tutorial: How to Create an STX App Using the STX Editor.

8. LiveArea Background Image

This is a full-screen image that appears in the LiveArea™, behind the **Start** image button. It is also used as a loading screen while the STX application is loading to launch on the PlayStation®Vita.

For more information, see Select Images in Chapter 4, Tutorial: How to Create an STX App Using the STX Editor.

9. In-app Image

This is a full-screen background image that is used in the STX application on the PlayStation®Vita. The STX installer renders the dialogs to the user on top of this image.

For more information, see Select Images in Chapter 4, Tutorial: How to Create an STX App Using the STX Editor.

10. The SoundTrack Table

The SoundTrack table shows all the tracks to be packaged and distributed in the PlayStation®Store through the STX application. Each row represents a track.

For more information, see Import Tracks and Albums and Edit Track Information in Chapter 4, Tutorial: How to Create an STX App Using the STX Editor.

11. Remove Tracks

This button removes all the tracks that have a selected checkbox from the SoundTrack table.

12. Param File Editor

This field enables you to create or select and then edit the `param_psp2.sfo` configuration file that is used by the PlayStation®Vita to set the application information.

For more information, see Select and Edit the Param File in Chapter 4, Tutorial: How to Create an STX App Using the STX Editor.

13. Build Package

This button represents the final step in the process of building a package for distribution. It builds the `.pkg` file according to the information provided in the tool.

For more information, see Build the Package in Chapter 4, Tutorial: How to Create an STX App Using the STX Editor.

3 STX Editor Project Structure

This chapter describes the structure of a project created in the SoundTrack eXporter Editor (STX Editor).

Overview

The data that you enter in the STX Editor is saved in a SoundTrack *project*. A SoundTrack project is composed of several files and directories. These are created when you save the project and during package building:

Project File (.json)

The project file contains the settings and details for the STX Editor tool. It also contains all the data, if any, that you have entered into the editor.

Param File (.sfo)

The param file specifies the configuration data used by the PlayStation®Vita STX application, such as the application title. You may choose the file path when saving a project or when referring to an existing file. An external tool, the **Param File Editor**, usually provided with the PlayStation®Vita SDK and also downloadable through the SDK Manager, enables you to edit the file. It is important to save the changes in this editor, as they will not be saved in the project file.

Application Package File (.pkg)

When in patch mode, the original distributable application package file (.pkg) needs to be specified, so that the patch build can identify the application to update.

Change Info File (.xml)

When in patch mode, a change info file needs to be created and specified. It can be modified with an update message that will be shown in the Update History option for the application on the PlayStation®Vita system.

Project Directory

This directory contains the project and configuration files, as listed in Table 1.

Output Directory

This is the directory where the final package directory will be placed. The package directory is called FullPackagePSVita if an application is built, or PatchPackagePSVita if a patch is built. Both directories contain the .pkg file, an .xml file, and a .zip file with submission materials.

4 Tutorial: How to Create an STX App Using the STX Editor

This chapter consists of a tutorial that explains the process of creating a SoundTrack project from scratch and building a package.

> **Note:** Some processes and applications, for example the **Param File Editor** for editing the configuration file, do not support directory names that contain space characters; therefore you should ensure that no filename or directory path contains spaces.

Launch the Tool

To launch the tool, double-click on the file STXEditor.exe. The editor opens, ready for you to create a new project. You can now provide the data as described in the following sections. However, no project files or directories are created at this stage.

Clicking **File > New** or **File > Close** will also present an empty UI, ready for you to create a new project.

Provide the Content ID and Passcode

The Content ID can be decided by a set of rules. For details, refer to the *Publisher Tool Overview* that is provided with the PlayStation®Vita SDK.

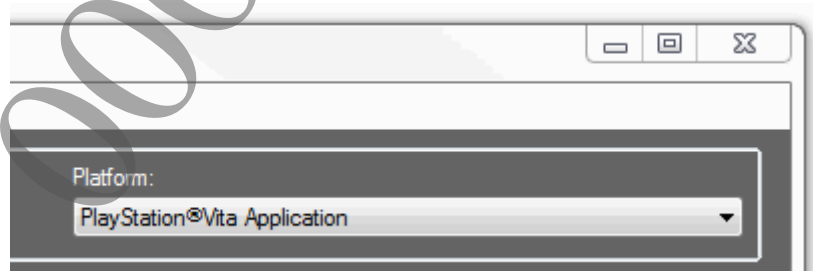To generate the passcode, click the **Generate** button next to the **Passcode** text field.

Figure 2   Passcode Field
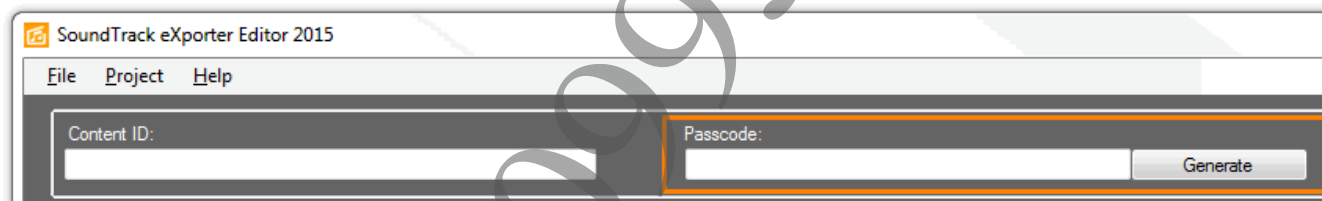
Choose a Platform

Select 'PlayStation®Vita Application' from the **Platform** dropdown list.

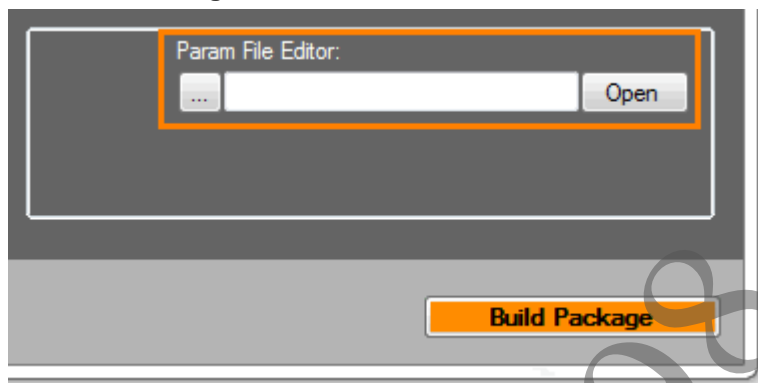**Figure 3   Platform Dropdown List**

Select and Edit the Param File

(1) If you already have a PlayStation®Vita `.sfo` file, select the file by clicking the search button ▢ next to the **Param File Editor** text field, and then click **Open**. Otherwise, go to step 2.
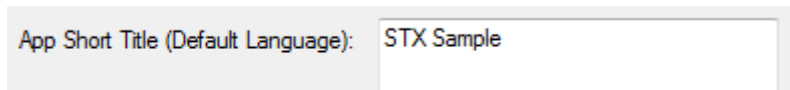
Figure 4   Param File Editor Field

(2) Click the **Open** button. Click **Yes** when asked to create a param file; if the project has not been saved, you will be prompted to save it, before a param file is generated in the project directory.

The param file launches in the external **Param File Editor for PlayStation®Vita**.

(3) Open the three tabs highlighted in Figure 5 and set the data as described below:

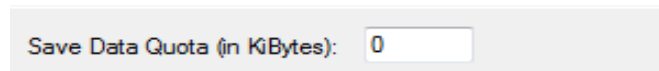**Figure 5   PlayStation®Vita Param File Editor**
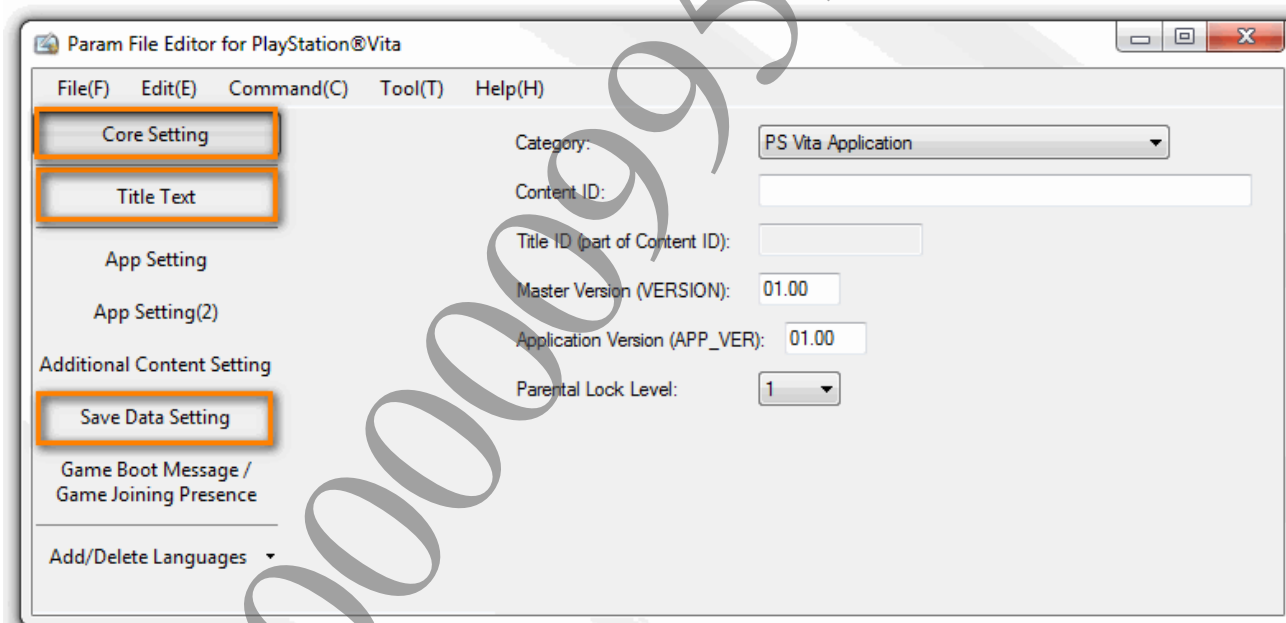
- **Core Setting**:
    - Ensure the **Category** is set to "PS Vita Application".
    - Ensure the **Content ID** is the same as in the STX Editor.
    - Ensure the **Parental Lock Level** is the same as for the related game.
- **Title Text**: Set the STX **App Short Title**.

Figure 6   App Short Title Field

App Short Title (Default Language):   STX Sample

- **Save Data Setting**: Set the minimum required memory to allocate to the STX application.

Figure 7   Save Data Quota Field

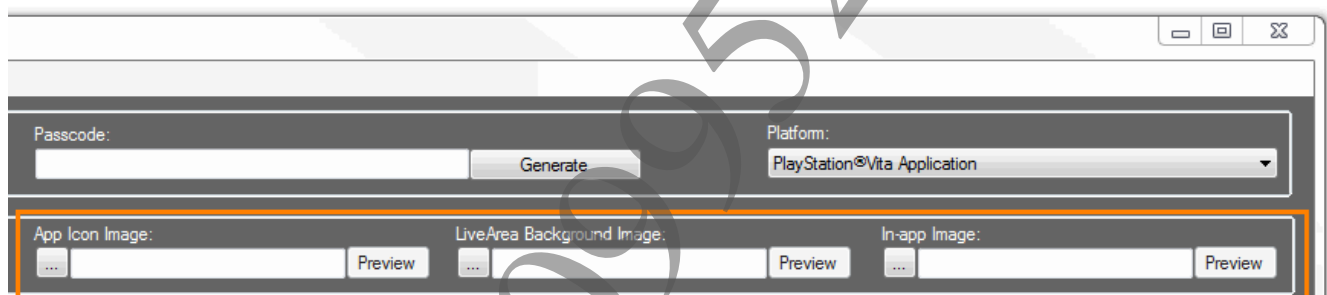Save Data Quota (in KiBytes):   0

(4) Set any other data as required.

Select Images

Three images must be selected for the STX application to work:

- **App Icon Image**: a 128x128px `.png` file (experienced PlayStation®Vita developers will recognize this image as the `icon0.png`).
- **LiveArea Background Image:** a 960x544px `.png`.
- **In-app Image**: a 960x544px `.png` file.

To select an image, click the search button [...] next to the image path's (initially empty) text fields.

Figure 8   STX Application Images

The images may be the wrong size. To preview an image in its original state, click the **Preview** button next to the image text field. In the final package-building stage, you can choose to let the application resize images to the right size.

> **Note:** When building, the application automatically creates 8BPP indexed copies of the original icon and LiveArea™ background images, to comply with the PlayStation®Vita requirements. Therefore, it is advisable to select images that work well with such compression by using a 256-colour palette.

Import Tracks and Albums

To import tracks into the SoundTrack table:

(1) Click the **Import Tracks…** button. An `.mp3` file browser is opened.

(2) Choose one or multiple `.mp3` files to import.

Alternatively, click the **Import Album…** button and import all the `.mp3` files from any chosen directory and its subdirectories.

Edit Track Information

You can now edit the track information by clicking on a cell on the targeted track's row in the SoundTrack table. Editing the track's information does not affect the actual music file; instead your changes are saved in the new copy of the music file that is created when you build the project.

Figure 9   Columns in the SoundTrack Table

Editable columns:

- **P (Checkbox):** [When in Patch mode] Include all tracks that are already in previous builds, and tick this checkbox so they won't be re-imported.
- **#** : A numerical field that determines the track order when the package is installed on the PlayStation®Vita. To order tracks as you wish them to be presented in the **Music** folder on the PlayStation®Vita, enter ascending numbers, starting with 1, in this column.

  > **Note:** The lowest number must be 1; this sets the track in question as the first track in the album.

- **X (Checkbox)**: To remove a track, select the checkbox and then click the **Remove Tracks** button.
- **Trackname**: The track's title.
- **Artist**: The artist(s) performing the track. Separate multiple values with '**;**'.
- **Year**: The year the track was/will be published.
- **Publisher**: The track's publisher.
- **Genres**: The track's genre(s). Separate multiple values with '**;**'.
- **Album**: The album that the track belongs to. You can define different albums for the tracks.
- **Album Thumbnail**: A button that launches a browser for `.jpg` files, from which you can choose one image to represent the track's album thumbnail.

The columns **Filename**, which shows the path to the track file, and **Duration**, which shows how long the track is, are not editable.

Build the Package

This is the final step in building the package.

To start the package-building process, click the **Build Package** button. When the process finishes, a message shows the result, and the output directory is opened.

If the process fails, the message shows the errors that need to be fixed. The output directory does not contain the files and directories described in Chapter 3, STX Editor Project Structure, but instead contains a temporary directory that can be deleted.

If the process is successful, a success message is displayed and the output directory shows the files and directories described in Chapter 3, STX Editor Project Structure. The `.pkg` file found there can now be distributed.

5 Tutorial: How to Create an STX Patch Using the STX Editor

This chapter consists of a tutorial that builds upon Chapter 4, Tutorial: How to Create an STX App Using the STX Editor, but explains the process of building a patch package instead of an application package. Follow the tutorial in Chapter 4 to learn how to build an STX project from scratch and what the different settings are for. In this tutorial, only the differences between an application package and a patch package are explained.

Choose a Platform

Select 'PlayStation®Vita Patch' from the **Platform** dropdown list.

Figure 10   Platform Dropdown List
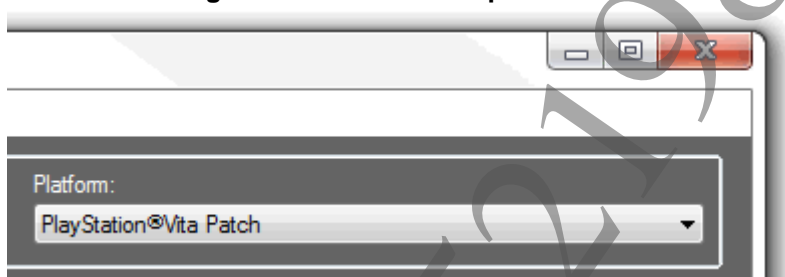
Reuse the Content ID and Passcode

The Content ID and passcode cannot be edited in this mode. They stay the same as the values entered in the PlayStation®Vita Application platform setting.

Select and Edit the Param File

(1) The process of selecting and creating a param file for a patch package is identical to that for an application package. If you already have a PlayStation®Vita `.sfo` file with patch settings, select it. Otherwise, go to step 2.

Figure 11   Param File Editor Field
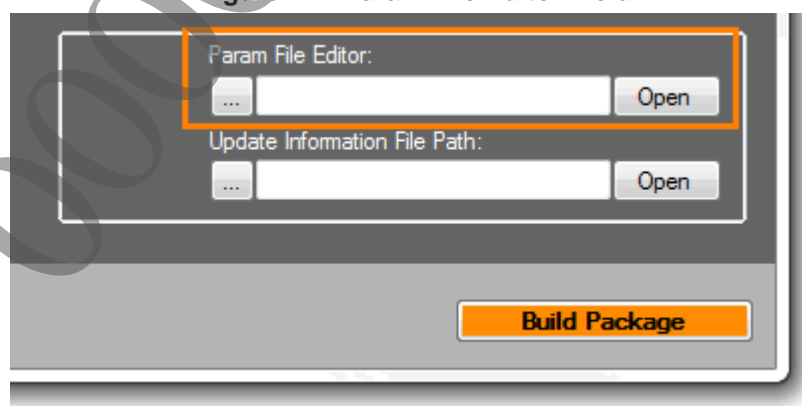
(2) Open the three tabs highlighted in Figure 12 and update the data as described below.

Figure 12   PlayStation®Vita Param File Editor
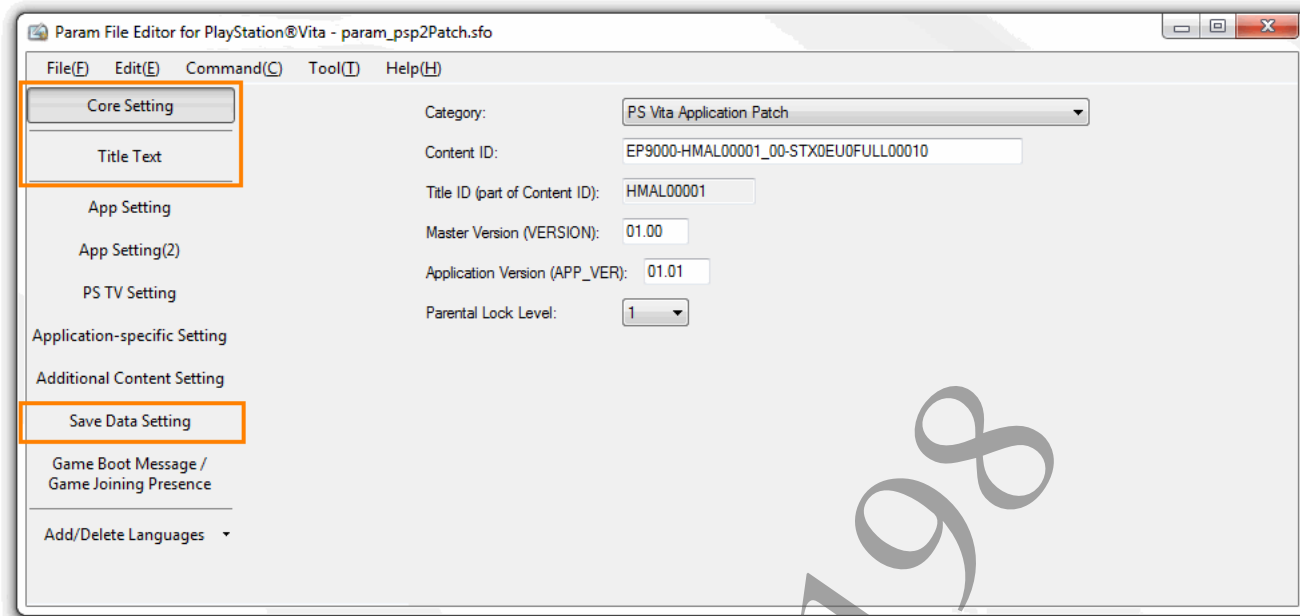
- **Core Setting**:
  - Ensure the **Category** is set to "PS Vita Application Patch".
  - Ensure the **Content ID** is the same as in the STX Editor.
  - Ensure the **Master Version** is updated
  - Ensure the **Application Version** is increased to a value where the first digit is not 0, for example '01.01'.
  - Ensure the **Parental Lock Level** is the same as for the related game.
- **Title Text**: Set the STX **App Short Title.** This can be the same title as for the original application.
- **Save Data Setting**: Update the minimum required memory to allocate to the STX application.
(1) Set any other data as required.

Update Images

You can select the same images as for the application package, or update the application with new icon and in-app background images. The LiveArea Background image cannot be updated.

Select Tracks and Albums to Patch

Import all tracks used in the application package and previous patches, so they can be referenced in this patch build, and also add new tracks for the patch package as usual. If you wish to delete tracks that were previously added, simply do not import them, or delete them from the project.

Edit Track Information

Exclude all existing tracks from being exported again, by ticking the **P** checkbox in each track field.

Select and Edit the Update Information File

The update information file contains release notes for a patch release and is used for displaying these notes on the device. With the update information, users can view the update content of all installed patches in the **Update History** screen that can be accessed from the home screen.

File Format and Size

- XML format text file
- UTF-8
- No BOM (Byte Order Mark)
- The linefeed code is LF
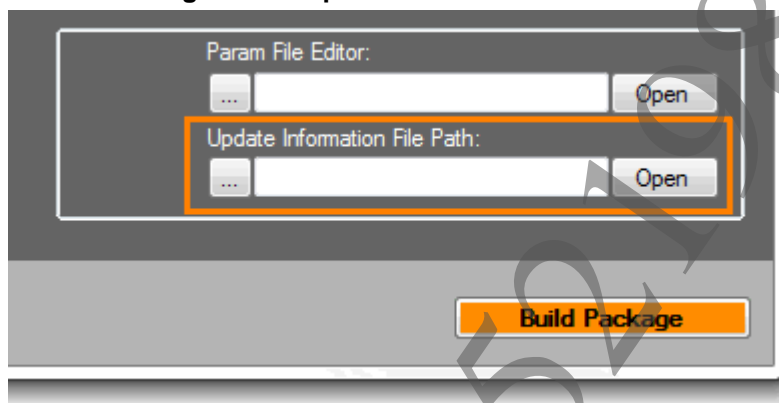- Maximum size if 64KiB

Selecting and Editing the File

(1) If you already have an update information file, open it for editing by clicking the search button next to the **Update Information File Path** field, and then clicking the **Open** button. Go to step 3.

Figure 13   Update Information File Path Field

If you do not have an update information file, go to step 2.

(2) Click the **Open** button next to the **Update Information File Path** field. When asked if a new file should be created, select **Yes**. A new file will be created according to a set template. Do not change the structure.

If you do not wish to include any release notes, close the launched editor without modifying the file, and skip step 3.

(3) Add the release notes:

Figure 14   Update Information File
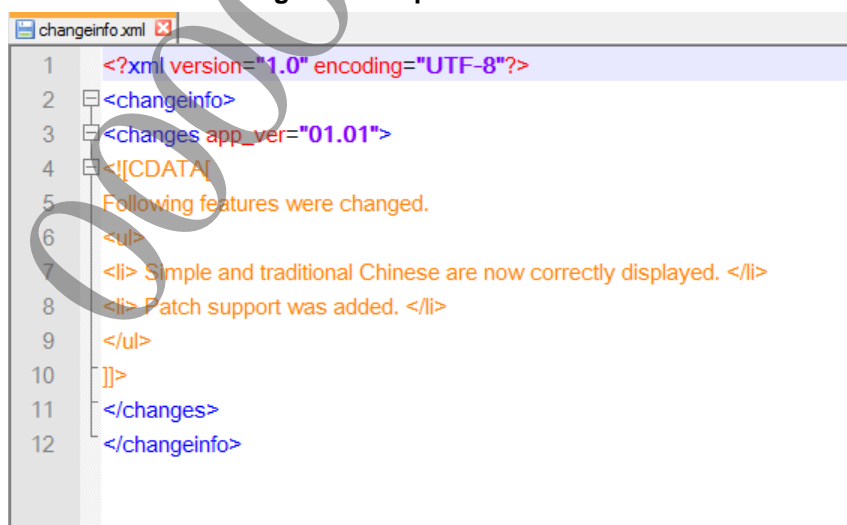
(a) In the `<changes app_ver="01.01">` element, make sure the patch version number is correct. This should match the APP_VER in the patch's `.sfo` file.

(b) In the CDATA[…] section, add any update information. If you wish, you can use the empty list provided in the template.

(c) Make sure the linefeed is LF, and then save and exit the program.

Application Package File

Reference the original application .pkg file here, by clicking the search button ⬚ next to the **Application Package File Path** field.

Build the Package

This is the final step in building the patch package. As in the application building process, click the **Build Package** button. When the process finishes, a message shows the result, and the output directory is opened. The distributable .pkg file can be in a folder called PatchPackagePSVita.

6 **PlayStation® TRC Requirements**

- The Parental Lock Level must match the level for the related game.

- A waiver request must be created on the PlayStation®Vita Developer Network for the following issue:

  - TRC R3146: Application uses a save file to determine if the content has previously been installed.