JPSXdec 0.96 (alpha) Manual

Michael Sabin
jpsxdec@gmail.com

Table of Contents

1 .About
1.1 ..... Java frameworks
2 .Gui
3 .Command-line
3.1 ..... Quick start
3.2 ..... Main command-line options
3.3 ..... Options for disc item types
4 .File formats recognized by jPSXdec
5 .Index files
6 .Replacing movies
6.1 ..... About the encoder
7 .Building from source
8 .License

1 About

jPSXdec is a PlayStation 1 media reader, decoder, and converter written using the Java framework.

Because jPSXdec uses the Java framework, it makes it almost universally cross-platform (including Windows, Mac, and Linux), but does require Java to be installed to work.

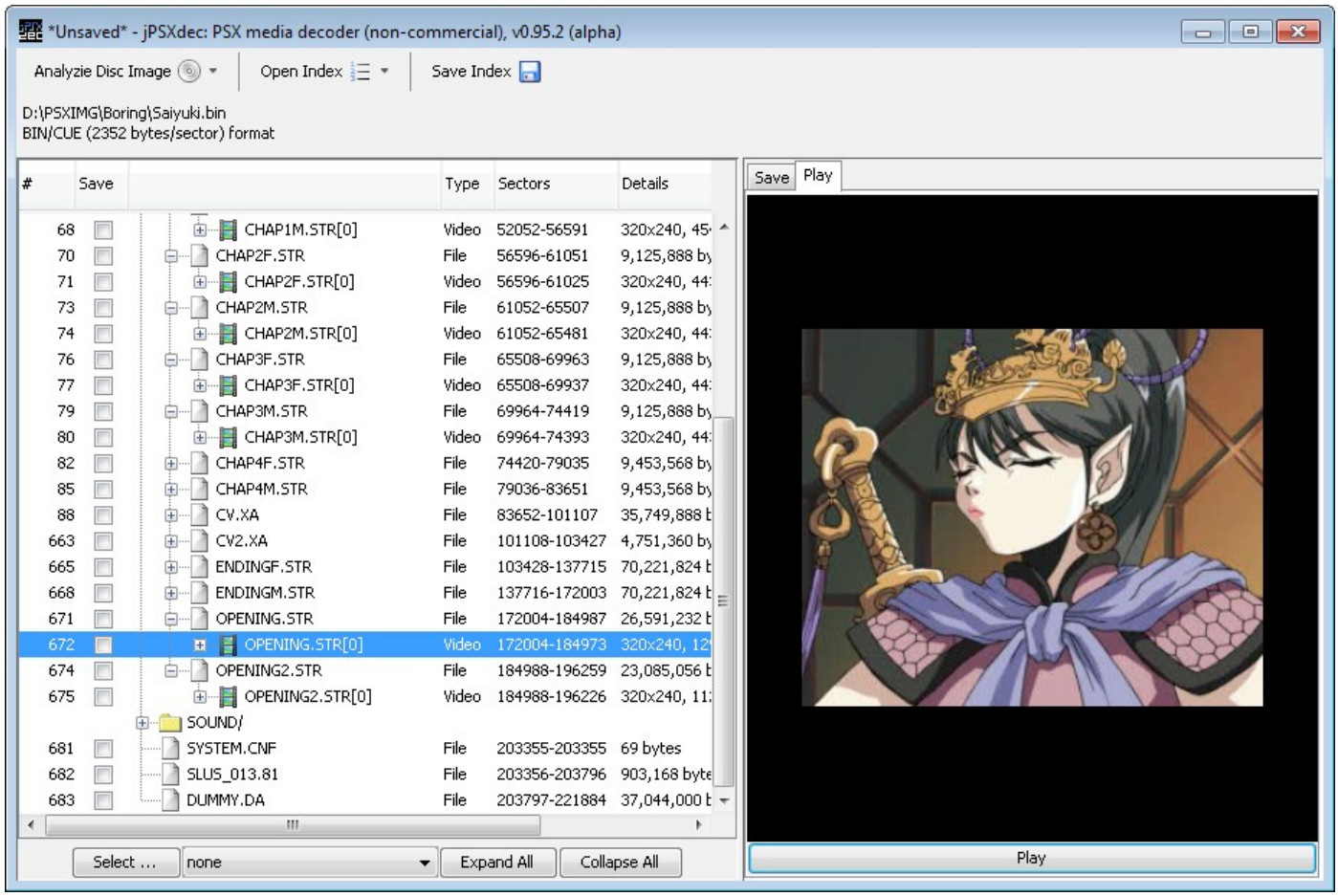
jPSXdec can read any file or image that comes from PlayStation discs, but it is recommended that the entire disc be ripped as a BIN/CUE file. You may have some success from ripping or copying individual files off the disc.

1.1 Java frameworks

There are several Java Virtual Machines and frameworks available. jPSXdec has only been tested on Oracle's Java.

2 Gui

Starting jPSXdec without argument will launch the graphical user interface.



2.1 Toolbar

TODO

2.2 List of items

TODO

2.3 Mass selection

TODO

2.4 Save panel

TODO

Save selected

TODO

2.5 Play panel

TODO

3 Command-line

3.1 Quick start

First the source file (whether it be a disc image or a single media file) needs to be analyzed and an index generated.
Example to generate an index:

```
java -jar jpsxdec.jar -f MyGameImage.bin -x MyGameImage.idx
```

Once the index is created, open it to review what was found in the source file.
Identify the number of an item you are interested in, and run the command to extract/decode it.
Example to decode the first (0th) item:

```
java -jar jpsxdec.jar -x MyGameImage.idx -i 0
```

3.2 Main command-line options

`java -jar jpsxdec.jar -?/-h/-help`
Displays main help.

`java -jar jpsxdec.jar -f <in_file> -x <index_file>`
Build an index of <in_file> and save it as <index_file>

`java -jar jpsxdec.jar [ -f <in_file> ] [ -x <index_file> ] <index_command> [index_command_options]`
Perform a command that requires an index. Either the <in_file> or <index_file> need to be specified, or both.
If only the `<in_file>` is specified, then it is indexed, but the index is not saved. This is useful for small .str files that only contain one or two streams.
If only the `<index_file>` is specified, then jPSXdec uses the file stored in the index as the <in_file>
If both the `<in_file>` and `<index_file>` are specified, then jPSXdec uses the index, but uses the `<in_file>` specified on the command-line instead of the file defined in the index.

The available <index_commands>:

-item/-i # [item_command] [item_command_options]
Process one disc item (see item's help for options)

-a <type> [item_command] [item_command_options]
Process all disc items of type audio, video, or file (see item's help for options)

Some shared [item_command]s:

If no command is given, then it performs the default decoding command

-play
Opens a very simple player to play movies and audio

-?/-h/-help
Displays detailed help about the item

-visualize <output_pdf>
Creates a massive pdf representing the physical layout the disc with color-coded sectors and index items

java -jar jpsxdec.jar -f <in_file> <in_file_command> [in_file_command_options]
Perform an operation on <in_file>

-copysect < # | #-# > <out_file>
Copy sectors to <out_file>

-sectordump <out_file>
Write list of sector types to <out_file> (for debugging)

~~-static <type>~~
~~Process special static file types~~

Additional option (optional):
-v/-verbose #
How much info to print: 0 for none, 1 for only errors, 2 for errors & warnings, 3 for normal, 4 for extra

3.3 Options for disc item types

jPSXdec can detect the following data on PlayStation discs.

STR Video

All types of PlayStation videos.
For the highest quality output, use these flags:

```
-quality high+ -vf avi:rgb
```

The PlayStation doesn't require videos to have a consistent frame rate. As such, some games have movies with variable frame rates, and most games have movies where the frame rate isn't precisely as detected by jPSXdec (but the variance isn't perceivable). When saving videos as an image sequence, frame rate is simply ignored. When saving as AVI, jPSXdec tries to place the frames in the AVI time-line as close to where they belong. This may mean that duplicate frames are inserted into the movie (thankfully AVI files have a convenient feature so duplication adds almost no extra size to files). Silent audio may also be inserted as needed in rare cases to keep the audio in sync. For more precise output, the `-preceiseav` and `-precisefps` options are available.

Command-line options

`-frameinfodump`
: Prints out detailed information about each video frame.

Options for decoding video items

`-dir <directory>`
: Specifies the output directory where item(s) will be saved. Default is the current directory.

`-quality <low, high, high+, psx>`
: Selects the decoding quality. Default is low. Option ignored if saving as demux or mdec format.

  `low` quality is very fast and uses integer based arithmetic.

  `high` is much slower, but should create better output because it uses floating point arithmetic (although it may not be noticeable without close inspection). It uses nearest-neighbor interpolation for chroma upsampling.

  `high+` is like `high`, but uses bilinear interpolation for chroma upsampling for a noticeable quality improvement. Not applicable when saving as `avi:yuv` or `avi:jyuv`.

  `psx` quality tries to duplicate what the PlayStation would generate. It's not perfect, but is currently better than any other decoder ever written.

`-vidfmt/-vf <format>`
: Selects the output format (default is `avi:mjpg`):

  `avi:mjpg` – Smallest output, but it is a lossy format so quality may be degraded. Use the `-jpg` option to set compression/quality level.

`avi:rgb` – Completely uncompressed, RGB (device independent bitmap, i.e. DIB) AVI. Very large, but lossless.

`avi:yuv` – Higher quality output, and smaller even than `avi:rgb`. Uses the fourcc YV12 codec.

`avi:jyuv` – Most similar to PlayStation video format. Uses the fourcc YV12 codec, but pixel values use the full [0-255] range (also known as "pc.601").

`png`, `jpg`, `bmp` – Saves a series of images. Use the `-jpg` option to set compression/quality level when saving `-vf jpg`.

`demux` – Demuxed compressed bitstream frame data found in separate sectors on the disc. 'Demuxing' is the first step in the decoding process, which simply combines the separate frame chunks into a contiguous block.

`mdec` – Generated from the demuxed bitstream data. This data would then be fed into the PlayStation MDEC chip, which would produce viewable images.

`-jpg <#>`

Selects the quality between 0 and 100 when saving a sequence of jpeg images, or as AVI with MJPG codec. Default is `75`.

`-frame <#> / -frames <#-#>`

Lets you specify that only 1 frame, or a range of frames should be decoded. Audio is ignored when using this option because accurate audio decoded must begin from the start of the audio stream (you can't start in the middle).

`-noaud`

Don't decode audio with the video (ignored when video does not have audio, or when not saving as AVI).

`-preciseav`

By default, when saving the audio and video as an AVI, they are set to start playing at precisely the same time. While that is probably how the video was intended to be seen, that is not exactly how it is played back on the PlayStation. The audio most likely starts where it first appears on the disc, which could be up to 0.15 seconds after the video starts. This option saves the audio and video to begin exactly when they appear on the disc (option ignored if not saving AVI with audio).

`-nocrop`

PlayStation 1 movies technically must have dimensions that are multiples of 16. In cases where games need movies that do not have such dimensions, the extra space is filled with garbage and is cropped off before being drawn on screen. When this option is used, that extra data is not cropped. Option ignored if saving as demux or mdec format.

`-ds <1 or 2>`

Movie frame rates are entirely dependent on the speed the disc spins.

PlayStation 1 can spin discs at 1x speed, or 2x speed. Nearly every game I've seen uses 2x speed for all their movies. jPSXdec can usually automatically detect the disc speed by analysis of parallel audio streams, but if the speed is unclear, it will assume 2x. In the rare case that is incorrect, this option lets you force using 1x disc speed instead (effectively halving the frame-rate). Option ignored if not saving as avi.

~~`-preciseFps`~~

~~Like the option above, the frame rate for most videos isn't exactly how the video is actually played back on the PlayStation. This option will save the AVI with frames appearing exactly how they are arranged on the disc, which will give the video a 75 or 150 fps rate (option ignored if not saving AVI with audio).~~

`-replace <str_replace_xml>`

Replaces frames of a video item according to the options in the xml file

XA Audio & Square Audio

Standard XA ADPCM audio used by Sony PlayStation 1 and Phillips CD-i, and audio from Square (now SquareEnix) games (e.g. Final Fantasy).

Command-line options

Options for decoding audio items

`-dir <directory>`

Specifies the output directory where item(s) will be saved. Default is the current directory.

`-vol <number between 0 and 100>`

This does more than just adjust the volume, because it does the adjustment before clamping the waveform and rounding it to an integer volume. Therefore the output should be better quality when scaling the volume at this point, and can even help in cases where a game's audio has distortions due to the clamping. Default is 100.

`-audfmt / -af <format>`

Output audio format. Possible options are `wav`, `aif`, `au`, and `snd`. Default is `wav`.

File (ISO9660)

Normal files on disc images are detected and can be extracted and saved. Note that jPSXdec ignores the file size and just saves the full sectors.

Command-line options

Options for saving file items

`-dir <directory>`

Specifies the output directory where item(s) will be saved. Default is the current directory.

`-iso`

jPSXdec can detect 4 different styles of disc images (raw+subchannel 2448 bytes/sector, raw 2352 bytes/sector, special 2336 bytes/sector, and iso 2048 bytes/sector). By default, files from these disc images are copied out with the same style as the source disc image. With this `-iso` option, all raw sector headers are ignored and the file is saved as if it had been normally copied off a disc. Care must be taken that this option isn't used on files containing XA audio streams, as that raw header information is required for decoding them.

TIM image

TIM images regularly used in PlayStation games.

Command-line options

`-imgfmt / -if <image format>`

Format to save the TIM image. Use item's help for available options. These options will be different depending on if the TIM image has a palette. Default is `png`.

`-pal <palettes to save>`

TIM images may come with one or more palettes. By default, all palettes are saved as a separate output files. This option lets you select exactly which palettes to save using a comma delimited list, or hyphens (no spaces). For example, to save the first and third, fourth and firth palettes, you could use this:

```
-pal 1,3-5
```

4 File formats recognized by jPSXdec

jPSXdec can read and understand two main file formats: disc images in the form of ISO and BIN/CUE. Since it can read BIN/CUE disc images, it can also read .STR and .XA files because those files are simply extracted portions of BIN/CUE disc images.

5 Index files

Index files contain important information about the contents of a disc image or media file. jPSXdec needs the index before it can perform any operations on a source file.

Index files may be edited by hand. If there's something wrong, jPSXdec will let you know.

The first line of index files contain a header indicating the jPSXdec version used to save the

file. jPSXdec won't open index files saved with a different version.

Index files contain a path to the source file that the index was created from. jPSXdec can use that so it doesn't always have to be respecified on the command-line.

All audio and video streams are listed separately, even if they should probably be combined. Actions upon video streams will also consider any parallel audio streams.

6 Replacing movies

Using the `-replace` option on video items, jPSXdec can re-encode and replace video frames within streaming movies on the disc or movie file. To use this feature, you must create a .xml file that tells jPSXdec what frames you want replaced, and how to replace them. The format is as follows:

```
<?xml version="1.0"?>
<str-replace version="0.1">

    <replace frame="10">newframe10.bmp</replace>

    <replace frame="13" format="mdec">newframe13.mdec</replace>

    <partial-replace frame="17" tolerance="5" mask="test.png"
rect="20,15,200,150">
        newpartialframe.png
    </partial-replace>

</str-replace>
```

The root `<str-replace>` tag needs the `version` attribute to equal the current file format version (0.1).

Within the `<str-replace>` tag, only `<replace>` and `<partial-replace>` tags are allowed.

The `<replace>` tag specifies that a frame should be completely replaced. The `frame` attribute is required which specifies the frame number to replace (the frame number starts with the first frame number as listed in the index file, which is usually 1).

The tag has one optional `format` attribute that can have the value of either `"bs"` or `"mdec"` which indicate that the replacement image isn't a normal image file but a 'bitstream' or 'mdec' file.

Within the `<replace>` tag is the file name of the image that should be encoded and inserted into the movie. Only .bmp and .png images are supported, unless the `format` attribute is used.

The `<partial-replace>` tag specifies that a frame should only be partially replaced. This is the perfect option for adding subtitles or other partial adjustments to a frame. The rest of the frame remains unchanged, so the quality loss due to re-encoding is minimized. This tag also has the required `frame` attribute. jPSXdec automatically compares the existing frame in the movie with the new image to detect what parts are different. It then has three optional

attributes that let you specify extra pixels to ignore, even if they are different.

The `tolerance` attribute is similar to the 'tolerance' option for the wand selection or fill tools of advanced image editors. When detecting what pixels have changed, jPSXdec will ignore different pixels if they are only different by the tolerance amount (on each RGB channel). The default value of this attribute is 0.

The `mask` attribute lets you specify an image that will indicate exactly what pixels to check for differences. Any pixel that is solid black in the mask image will not be compared. Non-black pixels will be compared according to the `tolerance` option.

The `rect` attribute is similar to the `mask` option. It lets you specify a simple rectangle region to compare. Pixels outside the region are ignored and considered unchanged. The value of this attribute is four numbers separated by commas: x, y, width, height. Pixels within the rectangle are compared according to the tolerance option.

Within the `<partial-replace>` tag is the name of the image that will partially replace the frame. Only .bmp and .png files are supported. Note that if no differences are detected between the frames, nothing is changed in the original.

6.1 About the encoder

The jSPXdec video encoder uses double-precision floating-point math. Unfortunately that's where the quality ends. It is very straight forward, and **doesn't** include any of the following features I wish it did.

- Smart chroma sub-sampling calculation
- Pre-anti-aliasing and Pre-blockiness reduction
- Temporal spreading of quantization error
- Shorter VLC adjustment searching

If there are any insights you could provide about these very awesome enhancements, please let me know.

7 Building from source

jPSXdec can be built using the readily available Apache Ant build system. An Ant build script has been included. If Ant is installed and configured properly, simply running the `ant` command in the directory with build.xml will generate a directory containing the binary program and everything needed to distribute it.

8 License

jPSXdec is licensed under a non-commercial license. See LICENSE.txt for details. If you would like to use jPSXdec in a way that is incompatible with this license, let me know. I am quite willing to make exceptions on a case-by-case bases. If you think your request is reasonable, I probably will too.