

JSI フォーマット仕様書

Ver. 1.00

内容

JSI フォーマットについて	1
基本データの規約.....	2
ファイル構造.....	3
識別コード.....	3
ヘッダー	3
※サムネイルの詳細.....	4
Short-TLV ブロック、Long-TLV ブロック	5
Short-TLV レコードのタイプ	7
Type=0x01 : タイトル	7
Type=0x02 : サブタイトル	7
Type=0x03 : 著者名.....	7
Type=0x04 : 出版社名	7
Type=0x05 : サークル名.....	8
Type=0x06 : 出版日（初版）	8
Type=0x07 : ユーザー定義属性.....	8
Long-TLV レコードのタイプ	9
Type=0x80 : ページ画像（JPEG）	9
Type=0x81 : ページ画像（PNG）	9
Type=0x82 : ページ画像（BMP）	9

JSI フォーマットについて

JSI フォーマットは、digi-PROVE により独自に開発されたファイルフォーマットです。ドキュメントスキャナーなどでスキャンした画像ファイルをまとめてひとつの文書ファイルにすることを目的としています。ファイル名の拡張子は、”JSI”とします。

JSI フォーマット自体は、ライセンスフリーとします。フリーソフト、商用ソフトを問わず、自由に使用できます。但し、勝手に仕様を変更したり、拡張したりすることは認められません。仕様変更や拡張をしたい場合は、別の名称、及び別の拡張子のフォーマットとしてください。しおり情報などを追加したい場合は、別のファイルにしてください。

同梱の JSIDoc.bmp、JSIDoc.ico も、JSI ファイルであることを示す目的であれば、自由に使用できます。

本フォーマットを使用したことにより障害や損害が発生した場合でも、当方は一切責任を負いません。

基本データの規約

- 16bit や 32bit の数値は、Little Endian とします。最下位 bit を bit0 と表現します。
- 文字列データは、UTF-16LE (Unicode) とします。ファイル内のデータとしては、終端の 0 は含みません。
- 文字列データには、タブや改行、及びその他の制御コードを含まないこととします。

ファイル構造

JSI フォーマットのファイル構造は、以下のようになっています。ファイルの先頭に識別コードがあり、その後にヘッダー、Short-TLV ブロック、Long-TLV ブロックが続きます。(Short-TLV ブロックは、存在しないこともあります。)



識別コード

識別コードは、JSI フォーマットであることを示す 4 バイトのデータです。

項目	長さ	説明
識別コード	4 Byte	0x05, 0x4A, 0x53, 0x49 C 言語の文字列で表現すると、“¥x05JSI”です (1 文字 8bit で、終端の 0 は除く)。

ヘッダー

ヘッダーには、文書ファイルの基本情報が含まれます。

項目	長さ	説明
ページ数	16 bit	ファイルに含まれるページの画像数(サムネイルを除く)と一致していること。
文書形式	8 bit	アプリケーションで文書として表示する形式を表します。 bit0: 0 = 表紙なし 1 = 表紙あり bit1: 0 = 左綴じ 1 = 右綴じ bit2: 0 = 単ページ表示 1 = 見開き表示 その他の bit は 0 とします。単ページ表示の場合は、表紙あり／なし、右綴じ／左綴じは意味を持ちません。 ビューアーにおいて、見開き表示のものをユーザーの操作で一時的に単ページ表示で提示することは可能とします。
オプション指定	8 bit	bit0: 0 = サムネイルなし 1 = サムネイルあり その他の bit は 0 とします。
サムネイル (オプション)	96000 Byte	オプション指定で、サムネイルありとなっている場合のみ存在します。 ※詳細は、後述します。

※サムネイルの詳細

サムネイルは、最初のページの画像を規定サイズにリサイズした画像とします。

- 幅 160、高さ 200 ピクセルのビットマップデータとします。
- 画像の一番下のラインから、上へ向かってデータを並べます。
- 1 ラインのピクセルデータは、左から右へ並べます。
- 1 ピクセルは BGR 順(各 8bit)の 24bit データとします。
- 元の画像の縦横比は、できるだけ変えないこととします。
- 余白が生じた場合、余白は黒とします。

データの形式としては、Windows の一般的な BMP ファイル(24bit カラー)のヘッダーを除いた画像データと同じです。

アプリケーションで表示する場合、必要に応じて縮小、拡大するものとします。(決まったサイズで表示することを強制するものではありません。)

<補足>

縦長なのは、横長の本に比べて縦長の本が圧倒的に多数だからです。

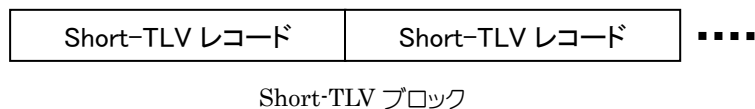
サムネイルとしてはサイズが大きすぎると思われるかもしれませんが、4K モニターや 4K テレビが普及しつつある状況を鑑みて大きめにしました。

JPEG などの圧縮形式でないのは、リスト表示を行う場合などの表示スピードを優先するためです。(ファイルを読み出すスピードと圧縮データをデコードするスピードのどちらが速いかはシステムにより異なるので、一概にどちらが良いとは言えない、というツッコミは無視します。)

Short-TLV ブロック、Long-TLV ブロック

Short-TLV ブロックは、任意の数の Short-TLV レコードの並びです (存在しないこともあります)。

Long-TLV ブロックは、一つ以上の Long-TLV レコードの並びです。



TLV レコードは、Type (データのタイプ)、Length (データの長さ)、Value (データ本体) から成ります。



Type: 1 Byte の値で、Value に含まれるデータの型を表します。

Short-TLV では、0x00～0x7F の範囲の値です。

Long-TLV では、0x80～0xFF の範囲の値です。

Length: Value の長さ (バイト数) を表します。

Short-TLV では 16bit です。

Long-TLV では 32bit です。

Value: Type により識別されるデータです。

※Length=0 で Value が無い状態は禁止とします。

TLV レコードは、原則として現れる順序を問いませんが、以下のような制約があります。

- Short-TLV ブロック内に Long-TLV レコードが存在してはいけません。
- Long-TLV ブロック内に Short-TLV レコードが存在してはいけません。
- アプリケーションは、規定されていない Type 値の TLV レコードは無視することとします。
- プロパティとして表示される同じ Type 値のデータ (例えば著者名) が複数存在する場合、プロパティで表示される順で現れることとします。
- ページの画像データについては、ページ順で現れることとします。

一つの Long-TLV レコードに、1 ページ(一つの画像データ)が格納されます。

JPEG、PNG、BMP 形式のファイルを、そのまま Value として格納します。Long-TLV ブロック内に、JPEG、PNG、BMP のページが混在できます。例えば文字主体のページは JPEG、挿絵や写真のページは PNG や BMP といった構成も可能です。

フォーマット	画質	サイズ	デコード速度
JPEG	劣化あり 但し品質の指定が可能で、高品質だと サイズが大きくなり、低品質だとサイ ズが小さくなります。	小～中	中程度
PNG	劣化なし	中	低速
BMP	劣化なし	大	高速

Short-TLV レコードのタイプ

Short-TLV レコードの存在は任意なので、Short-TLV ブロックが存在しないこともあります。

Type=0x01: タイトル

ファイルに、たかだか1回現れることとします(無しも可)。

Length(16bit) = 文字列の長さ(バイト数)

Value: タイトルの文字列(127 文字以内とすること)

Type=0x02: サブタイトル

ファイルに、たかだか1回現れることとします(無しも可)。タイトル無しで、サブタイトル有りは許容されますが、アプリケーションによっては表示されないかもしれません。

Length(16bit) = 文字列の長さ(バイト数)

Value: サブタイトルの文字列(127 文字以内とすること)

Type=0x03: 著者名

ファイルに、任意の回数現れることができます(無しも可)。一つの TLV レコードに一人の名前を含むものとします。

Length(16bit) = 文字列の長さ(バイト数)

Value: 著者名の文字列(127 文字以内とすること)

Type=0x04: 出版社名

ファイルに、任意の回数現れることができます(無しも可)。一つの TLV レコードに一社の名前を含むものとします。

Length(16bit) = 文字列の長さ(バイト数)

Value: 出版社名の文字列(127 文字以内とすること)

Type=0x05: サークル名

ファイルに、任意の回数現れることができます(無しも可)。一つの TLV レコードに一つの名前を含むものとします。

Length(16bit) = 文字列の長さ(バイト数)

Value: サークル名の文字列(127 文字以内とすること)

Type=0x06: 出版日(初版)

ファイルに、たかだか1回現れることとします(無しも可)。

Length(16bit) = 4

Value:

年(16bit)	西暦の年
月(8bit)	月(不明な場合は 0 とします)
日(8bit)	日(不明な場合は 0 とします)

Type=0x07: ユーザー定義属性

ファイルに、任意の回数現れることができます(無しも可)。編集者名や印刷所名、第二版以降の版数と発行日、メモなどを記録したい場合に利用できます。(第二版以降の発行日などは、文字列で表現してください。ちなみに、初版の出版日を数値にしているのは、出版日での並べ替えをしやすいするためです。)

Length(16bit) = Value の長さ

Value:

- 属性名の長さ(8bit、バイト数)
- 属性名の文字列(16 文字以内とすること)
- 属性の長さ(8bit、バイト数)
- 属性の文字列(127 文字以内とすること)

Long-TLV レコードのタイプ

Long-TLV ブロックには、最低でもどれか一つの Long-TLV レコードが存在することとします。

Type=0x80: ページ画像(JPEG)

Length(32bit) = JPEG データの長さ

Value: JPEG 画像ファイルのデータ

Type=0x81: ページ画像(PNG)

Length(32bit) = PNG データの長さ

Value: PNG 画像ファイルのデータ

Type=0x82: ページ画像(BMP)

Length(32bit) = BMP データの長さ

Value: BMP 画像ファイルのデータ

奥付

Ver.1.00 2021 年 1 月 23 日

本書は、著作権法上の保護を受けています。

本書はフリーソフトの一部であり、digi-PROVE が作成したアーカイブの形態での複製、配布は自由に行うことができます。また、プログラムをインストールしたPC内部、およびそのバックアップ目的での記憶メディアへの保存に限り複製を行うことができます。以上の形態以外の形態では、本書の一部または全部について、digi-PROVE から文書による許諾を得ずに、いかなる方法によっても無断で複写、複製することは禁じられています。

Copyright© 2021 digi-PROVE

メールアドレス: digiprove@outlook.jp