Chapter 15

PDF and JSON

```
cpdf in.pdf -output-json -o out.json
     [-output-json-parse-content-streams]
     [-output-json-no-stream-data]
     [-output-json-decompress-streams]
     [-output-json-clean-strings]
cpdf -j in.json -o out.pdf
```

In addition to reading and writing PDF files in the original Adobe format, `cpdf` can read and write them in its own CPDFJSON format, for somewhat easier extraction of information, modification of PDF files, and so on.

15.1 Converting PDF to JSON

We convert a PDF file to JSON format like this:

```
cpdf -output-json in.pdf -o out.json
```

The resultant JSON file is an array of arrays containing an object number followed by an object, one for each object in the file and two special ones:

- Object -1: CPDF's own data with the PDF version number, CPDF JSON format number, and flags used when writing (which may be required when reading):
  - `/CPDFJSONformatversion` (CPDFJSON integer (see below), currently 2)
  - `/CPDFJSONcontentparsed` (boolean, true if content streams have been parsed)
  - `/CPDFJSONstreamdataincluded` (boolean, true if stream data included. Cannot round-trip if false).
  - `/CPDFJSONmajorpdfversion` (CPDFJSON integer)

  - `/CPDFJSONminorpdfversion` (CPDFJSON integer)

- Object 0: The PDF's trailer dictionary
- Objects 1..n: The PDF's objects.

Objects are formatted thus:

- PDF arrays, dictionaries, booleans, and strings are the same as in JSON.
- Integers are written as `{"I": 0}`
- Floats are written as `{"F": 0.0}`
- Names are written as `{"N": "/Pages"}`
- Indirect references are integers
- Streams are `{"S": [dict, data]}`
- Strings are converted from UTF16BE/PDFDocEncoding to UTF8 before being encoded in JSON. This process is fully reversible: it is to allow easier editing of strings. This does not happen to strings within text operators in parsed content streams, nor to /ID values in the trailer dictionary, since neither is UTF16BE/PDFDocEncoding to begin with.

Here is an example of the output for a small PDF:

```
[
  [
    -1,
    { "/CPDFJSONformatversion": { "I": 2 },
      "/CPDFJSONcontentparsed": false,
      "/CPDFJSONstreamdataincluded": true,
      "/CPDFJSONmajorpdfversion": { "I": 1 },
      "/CPDFJSONminorpdfversion": { "I": 1 } }
  ],
  [
    0,
    { "/Size": { "I": 4 }, "/Root": 4,
      "/ID" : [ <elided>, <elided>] } ],
  [
    1, { "/Type": { "N": "/Pages" }, "/Kids": [ 3 ], "/Count": { "I": 1 } }
  ],
  [
    2,
    {"S": [{ "/Length": { "I": 49 } },
     "1 0 0 1 50 770 cm BT/F0 36 Tf(Hello, World!)Tj ET"] }
  ],
  [
    3, { "/Type": { "N": "/Page" }, "/Parent": 1,
    "/Resources": {
      "/Font": {
```

```
        "/F0": {
          "/Type": { "N": "/Font" },
          "/Subtype": { "N": "/Type1" },
          "/BaseFont": { "N": "/Times-Italic" }
        }
      }
    },
    "/MediaBox":
      [{ "I": 0 }, { "I": 0 },
       { "F": 595.2755905510001 }, { "F": 841.88976378 }],
    "/Rotate": { "I": 0 },
    "/Contents": [ 2 ] } ],
[
  4, { "/Type": { "N": "/Catalog" }, "/Pages": 1 } ]
]
```

The option `-output-json-parse-content-streams` will also convert content streams to JSON, so our example content stream will be expanded:

```
2, {
"S": [
  {}, [
  [
  { "F": 1.0 }, { "F": 0.0 }, { "F": 0.0 }, { "F": 1.0 }, { "F": 50.0 }, {
  "F": 770.0 }, "cm" ], [ "BT" ], [ "/F0", { "F": 36.0 }, "Tf" ], [
  "Hello, World!", "Tj" ], [ "ET" ] ]
] } ], [
```

The option `-output-json-no-stream-data` simply elides the stream data instead, leading to much smaller JSON files.

The option `-output-json-decompress-streams` keeps the streams intact, and decompresses them.

The option `-output-json-clean-strings` converts any UTF16BE strings with no high bytes to PDFDocEncoding prior to output, so that editing them is easier.

15.2 Converting JSON to PDF

We can load a JSON PDF file with the `-j` option in place of a PDF file anywhere in a normal `cpdf` command. A range may be applied, just like any other file.

```
cpdf -j in.json -o out.pdf
```

It is not required that `/Length` entries in CPDFJSON stream dictionaries be correctly updated when the JSON file is edited: `cpdf` will fix them when loading.