PyPDF file specification

Version 1.0

*2019-05-27 Dirk van den Bekerom*

**This is work in progress, specification is not yet finished. Please contact me at** [**dcmvdbekerom@gmail.com**](mailto:dcmvdbekerom@gmail.com) **if you have any questions.**

The PyPDF file is specification is an extension to the PDF file specification, that allows files to be read both as PDF file and as python file. It is designed for cases where a python script generates a PDF file, and the generating script is stored in the PDF file, in such a way that by reading the PDF file in a python editor lets you edit the generating code directly. By its very definition, the PyPDF file must be compliant to both PDF and Python file specifications and syntactically valid for both.

**File structure**

The PyPDF file structure is an extension of the PDF file structure. It has some additional sections/limitations to make it compatible with a Python interpreter.The file structure is as follows:

*Header*

*Body*

*Xref*

*Trailer*

**Header**

The Header line %PDF-V.V is preceded by a ‘#’ to make the line not read by Python interpreter. The Header is followed by a space ‘ ‘, immediately followed by the body (without a linespace in between). This way the object header for the embedded generating script will also be ignored by a Python interpreter.

**Body**

The file structure of the PyPDF body is identical to the PDF body with some restrictions:

* Non-ASCII characters other than whitespace are not allowed. This means that every stream object must ultimately be encoded by either ASCIIHexDecode or ASCII85Decode. The stream may be encoded by any other filter before either of these are finally used. The exception is the generating script (see next), which my not be encoded at all.
* The generating python script is added as an embedded. Its stream object must be the first object to appear in the PyPDF file. The stream object’s dictionary must be entirely void of linefeeds up to and including the word ‘stream’ indicating the start of the filestream. The stream for the generating script may not be encoded at all. The generating script is the only place in the PyPDF file where non-ASCII characters may be used. This ensures that the python interpreter can read the entire PyPDF file.
* When attaching the generating script to the PDF document, a new line and three double quotes ‘\n”””’ are appended to the generating file in order for the python interpreter to ignore the following PDF syntax. The addition of these four bytes (‘\n”””’) should be included in the length of the stream object.

**Xref**

The xref is identical to the PDF specification. It is worth noting that offsets in the xref table are relative to the first instance of ‘%PDF’ in the header. This means that in a PyPDF file there is a difference of 1 between the offsets in the xref and the actual address positions in the file, because of the appended ‘#’ in the beginning of the file.

**Trailer**

The trailer is identical to the PDF specification, with two additional lines:

* The line after the %%EOF line is a 10 digit zero padded number that specifies the total file length. This makes it easy for the python interpreter to determine if code was added to the generating script and allows for ammendmend of the xref table.
* The final line in the PyPDF file is three double quotes ‘”””’ to close the three double quotes started at the end of the generating file, so the python interpreter will ignore all PDF syntax.

It is a requirement that all files referenced in the generating script are embedded in the PyPDF file.