What packing a document saves

A PDF is mostly dictionaries — the catalogue, the page tree, every page, every font, every annotation — and a dictionary is not a stream, so each one is written out in full and none of them is compressed.
Object streams gather them into streams that are compressed together, and the table stating where every object lives becomes a stream too. The document a reader sees is the same one either way.

The document weighed here holds 24 pages of text, each with a heading of its own.

The same document, weighed twice

Spelled out	22 877 bytes
Packed	18 489 bytes
Saved	4 388 bytes

Packing this document saved 19 per cent of it. What it saves goes up with the number of pages, because a page costs a handful of dictionaries and dictionaries are what packing compresses.

This very file is packed, so a reader that opened it has read one of the two documents being weighed.

The same document, weighed other ways

With compression off it weighs 31 242 bytes; at level 1, 26 906; at level 6, the default, 22 877; at level 9, 22 866.

Given a title, it weighs 22 971 bytes with its information dictionary and 25 418 without it, the title then said in its metadata alone; the dictionary is written: no.

A dry run, which writes nothing, counts 24 pages and 56 objects, written as PDF 1.7 though PDF 1.4 would do, and 0 warnings.

Written with its warnings reported it weighs 22 877 bytes, and 22 877 written to a stream the same way; warnings: 0.

Packed into object streams: no; laid out first page first: no; the version it needs: PDF 1.4.

With a cover put in front and a bookmark to it put first, it holds 25 pages, and its first bookmark reads “Cover”.

A line of four hundred characters measured inside a measuring pass costs the file nothing: 7 602 bytes, and 7 602 with the pass held open by hand, against 24 979 when measured outside one.