



Processor



Linearize PDFs



PROCESSOR > GUIDES

Efficiently linearize PDFs on Linux with Document Engine



PSPDFKit Processor has been deprecated and replaced by [Document Engine](#). To migrate to Document Engine and unlock advanced document processing capabilities, refer to our migration guide. Learn more about these enhancements on our [blog](#).

You can linearize PDFs using PSPDFKit Processor.

Before you get started, make sure [Processor is up and running](#).

You can download and use either of the following sample documents for the examples in this guide:

✧ [Example eight-page PDF](#)

✧ [Example four-page PDF](#)

You'll be sending [multipart POST requests with instructions](#) to Processor's `/build` endpoint. To learn more about multipart requests, refer to our blog post on the topic, [A Brief Tour of Multipart Requests](#).

Check out the [API Reference](#) to learn more about the `/build` endpoint and all the actions you can perform on PDFs with PSPDFKit Processor.

PDF Linearization



ASK AI

A linearized PDF file is organized in a special way to enable efficient incremental access in a network environment. Enhanced viewer applications can recognize that a PDF file has been linearized and take advantage of that organization.

To linearize a PDF, use the following example:

SHELL HTTP

```
1 curl -X POST http://localhost:5000/api/build \
2   -F document=@/path/to/example-document.pdf \
3   -F instructions='{
4     "parts": [
5       {
6         "file": "document"
7       }
8     ],
9     "output": {
10      "type": "pdf",
11      "optimize": {
12        "linearize": true
13      }
14    }
15  }' \
16   -o result.pdf
```

Licensing

To linearize PDFs with PSPDFKit Processor, the Linearization feature needs to be included in your license. [Contact Sales](#) to add linearization to your license. After the new component is added to your license, update the offline `LICENSE_KEY` in your PSPDFKit Processor configuration.

Other Types of PDF Compression

You can perform both linearization and compression in a single request to `/build` if both features are enabled in your license:

```
1 instructions = {
2   ...
3   output: {
4     type: "pdf",
5     optimize: {
6       grayscaleText: true,
```

```
7     grayscaleGraphics: true,  
8     grayscaleFormFields: true,  
9     grayscaleAnnotations: true,  
10    disableImages: true,  
11    mrcCompression: true,  
12    imageOptimizationQuality: 2,  
13    linearize: true,  
14  }  
15 }  
16 }
```

To learn more about other types of compression supported by PSPDFKit Processor, refer to the [PDF compression guide](#).

Was this helpful?

✓ YES

✗ NO

Questions? [Contact us](#)

