A Softer Space DE, Inhaber Tom Moya Schiller | Heerstraße 3 b | 60488 Frankfurt am Main, Germany

Latest version of this document:
asofterspace.com/pdfQrReader/manual.pdf

Phone: +49 176 511 27307
Email: info@asofterspace.com
Website: asofterspace.com
VAT Nr: DE319451065

Documentation for:
PDF QR Reader
version 0.0.0.1(37)

PDF QR READER MANUAL

Date: 30th March 2019

The PDF QR Reader is a small program that reads a PDF file, finds QR codes inside of it, reads out their contents and reports the results in several different formats.

Contents

Default Setup

To set up the PDF QR Reader you can just download a zip file, call a setup script, and it should start working.

In particular, you can download the latest zip from:

http://asofterspace.com/pdfQrReader/pdfQrReader.zip

To run the PDF QR Reader on your computer, you need to install Java on it.
(Any JRE or JDK of Java 7 or higher will be sufficient.)

To check that you have Java installed, you can type in a terminal:

```
java -version
```

Once Java is found, unzip the PDF QR Reader file, e.g. under Linux by typing:

```
unzip pdfQrReader.zip
```

You now have the `run.bat` and `run.sh` commands available to start the program.
(See chapter "Basic Operations.")

Setup using Manual Build

If you want to build the PDF QR Reader yourself, then you first of all should ensure that you have a JDK (Java Development Kit) of Java 7 or higher on your machine.
You also have to ensure that `javac` is on the `PATH` and can be found.
To check both, you can type in a terminal:

`java -version` and `javac -version`

Once you have set up the JDK, clone the cdm repository onto your machine:

```
git clone https://github.com/ASofterSpace/PdfQrReader.git
```

Finally, build the latest version of the cdm commandline tool by calling under Windows:

```
build.bat
```

Or under Linux:

```
./build.sh
```

Should you want to create a zip release file of what you just built, then you have to get your hands on a Java 7 `rt.jar` (such that the released zip can be created in a way compatible with Java 7 and above.)
Once you have it, put this file at the location:

```
other/java7_rt.jar
```

Then, under Linux you can call:

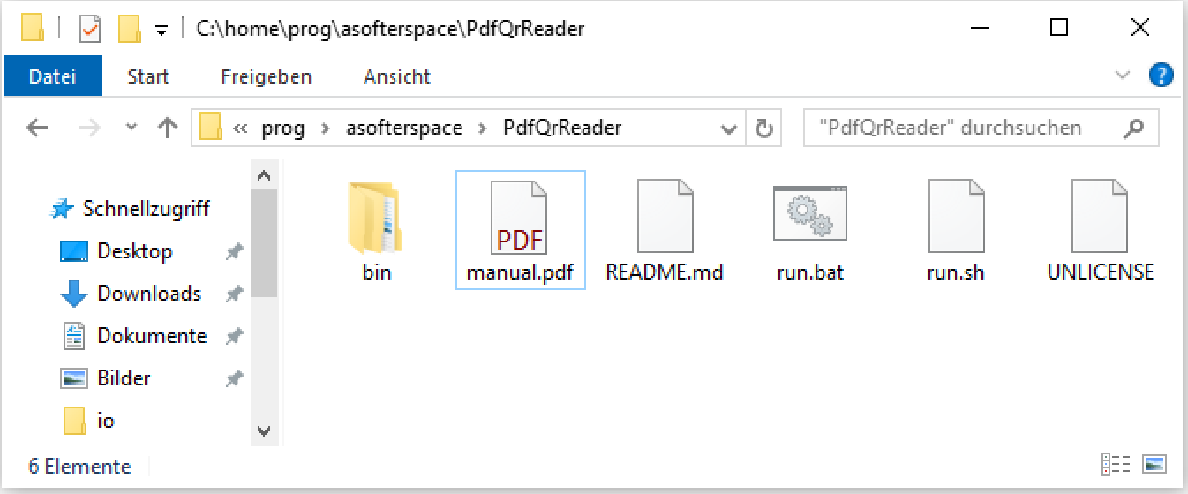
```
./release.sh
```

(from the same directory in which you also executed the `./build.sh` command.)

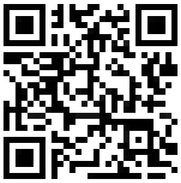
Basic Operations

Take a PDF file that contains pictures with QR Codes and put it into the same directory as the PDF QR Reader.

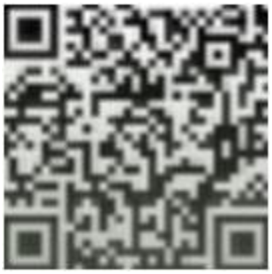
As an example, we will be using this manual which you are reading right now:



Inside the PDF, the QR codes should ideally be contained in their own picture elements, such as this one:



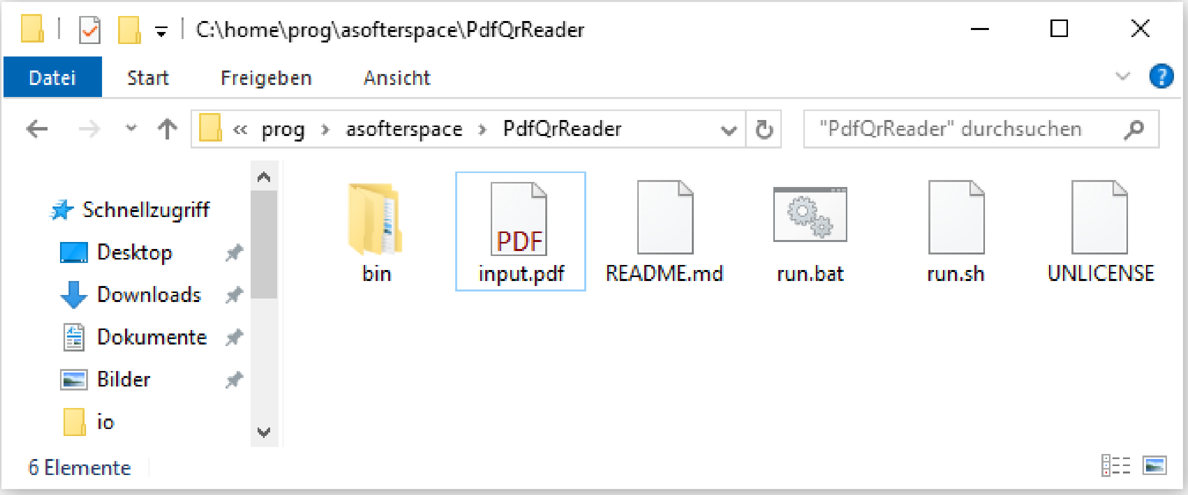
Even slightly corrupted QR codes should be recognized, such as this one, which is rotated and had its colors washed out a bit:



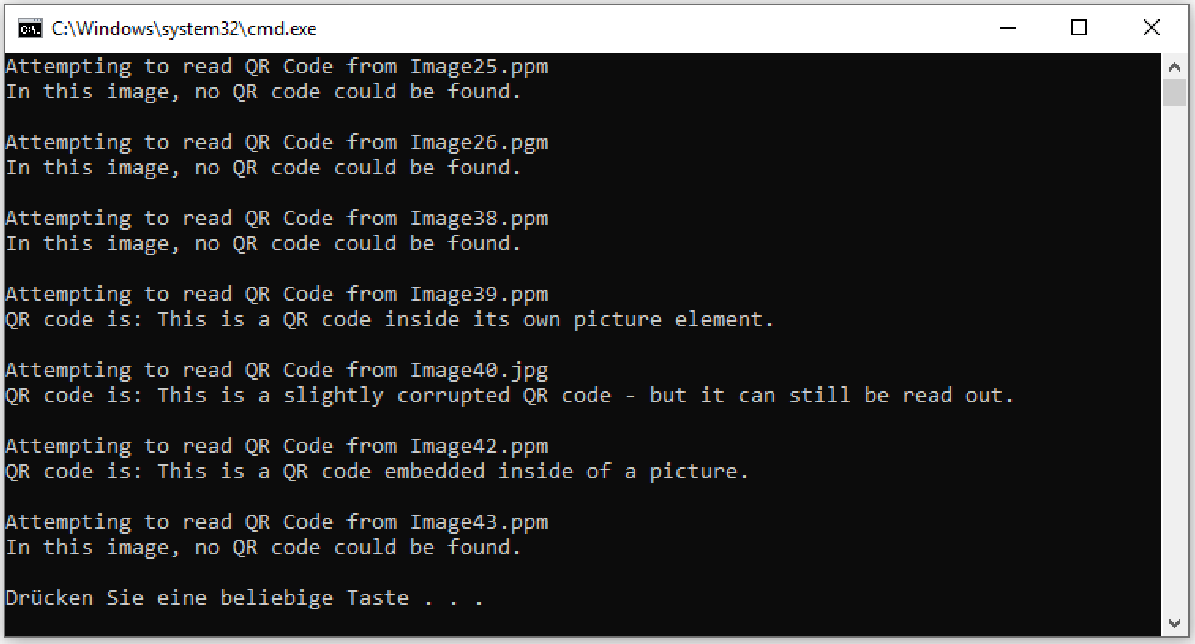
Finally, also QR codes embedded into other pictures can be recognized, but this is still a bit experimental – if you have a particular file for which this does not yet work, please let us know and we can probably improve this program for you:

a softer space

Now rename the PDF file to `input.pdf`:

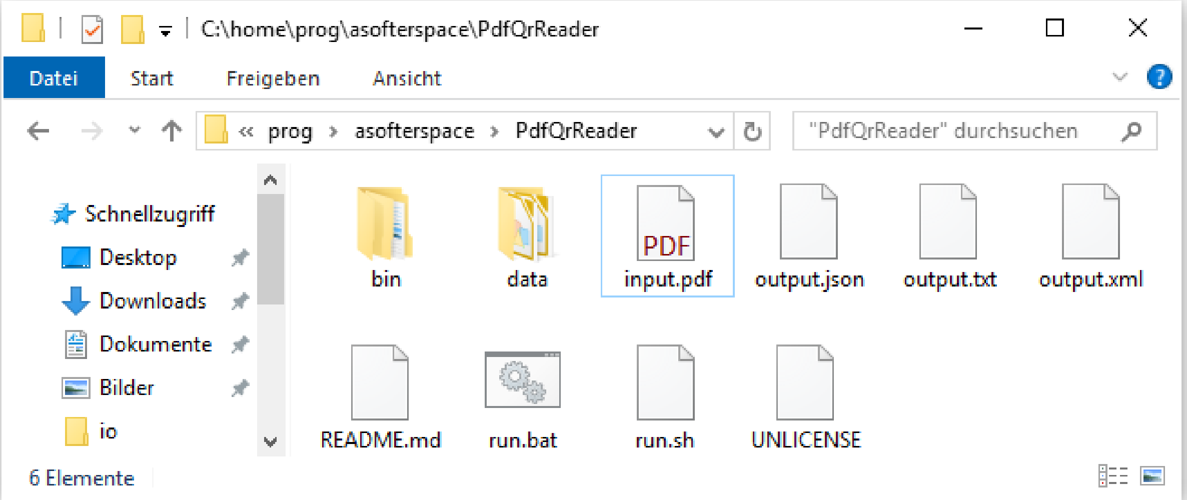


Run the PDF QR Reader by calling `run.bat` (on Windows) or `run.sh` (on Linux):



```
Attempting to read QR Code from Image25.ppm
In this image, no QR code could be found.

Attempting to read QR Code from Image26.pgm
In this image, no QR code could be found.

Attempting to read QR Code from Image38.ppm
In this image, no QR code could be found.

Attempting to read QR Code from Image39.ppm
QR code is: This is a QR code inside its own picture element.

Attempting to read QR Code from Image40.jpg
QR code is: This is a slightly corrupted QR code - but it can still be read out.

Attempting to read QR Code from Image42.ppm
QR code is: This is a QR code embedded inside of a picture.

Attempting to read QR Code from Image43.ppm
In this image, no QR code could be found.

Drücken Sie eine beliebige Taste . . .
```

Several output files will now have been created, such as JSON, TXT, XML:



The TXT output just contains one line for each QR code:

```
This is a QR code inside its own picture element.
This is a slightly corrupted QR code - but it can still be read out.
This is a QR code embedded inside of a picture.
```

The JSON output contains a `qrcodes` array:

```
{
  "qrcodes": [
    "This is a QR code inside its own picture element.",
    "This is a slightly corrupted QR code - but it can still be read out.",
    "This is a QR code embedded inside of a picture."
  ]
}
```

The XML output contains `qrcode` elements:

```
<?xml version="1.0" encoding="UTF-8"?>
<qrcodes>
  <qrcode>This is a QR code inside its own picture element.</qrcode>
  <qrcode>This is a slightly corrupted QR code - but it can still be read
out.</qrcode>
  <qrcode>This is a QR code embedded inside of a picture.</qrcode>
</qrcodes>
```

License and Support

We at A Softer Space really love the Unlicense, which pretty much allows anyone to do anything with this source code. For more info, see the file UNLICENSE in the git repository. If you desperately need to use this source code under a different license, contact moya@asofterspace.com - I am sure we can figure something out.

Finally, if you like this tool and would be happy for it to be developed further, for bugs that you find to be fixed, and even for new features that you have in mind to be added, then just let us know and we will offer you a service contract at such great conditions that you cannot possibly say no to it. :)

If it's stupid but it works,
it isn't stupid.
~ Murphy