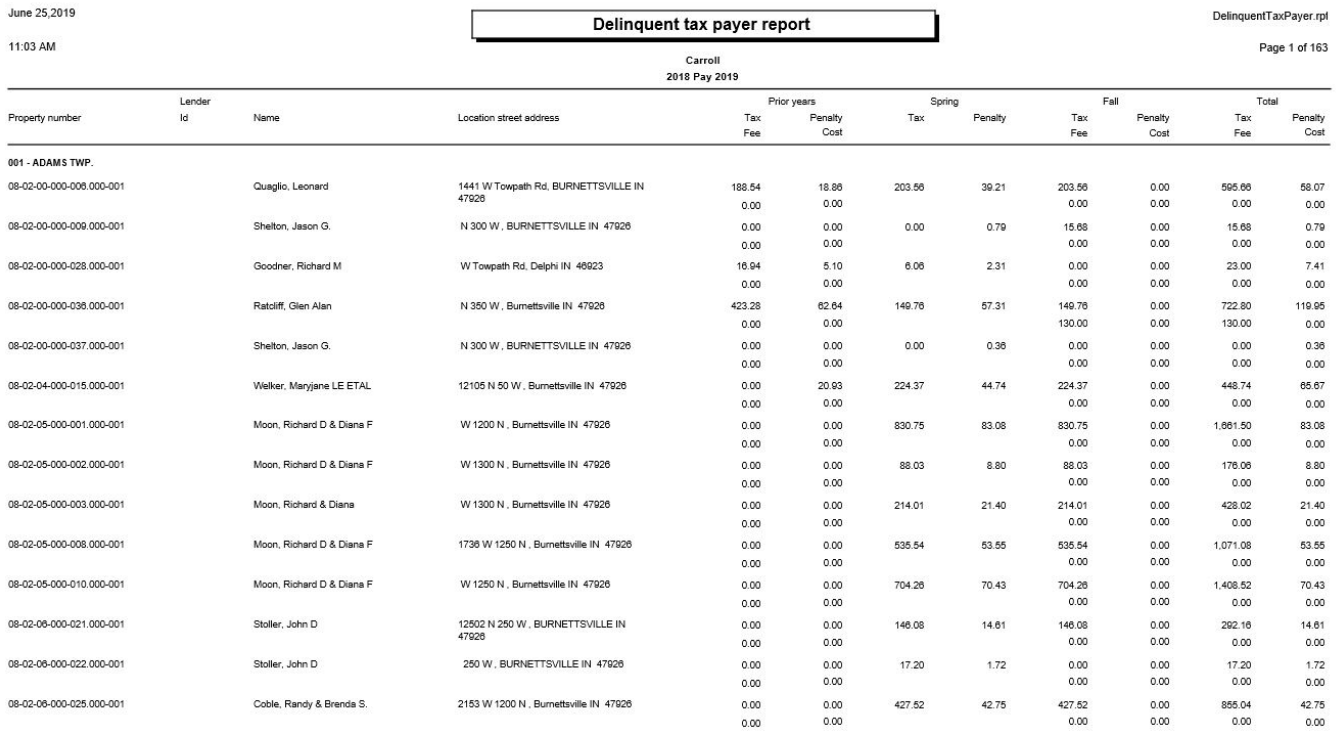
Extract PDF

Notes: This python scripts scrapes PDFs from a Delinquent tax payer report from different counties and creates a csv output file with the following items: **Property Number, Prior Years Tax, Prior Years Penalty, Spring Tax, Spring Penalty, Fall Tax, Fall Penalty, Total Tax, Total Penalty, Prior Years Fee, Prior Years Cost, Fall Tax Fee, Fall Penalty Cost, Total Tax Fee, Total Penalty Cost**. The filename is unique and is based on the county name listed at the top of the PDF. The output file is named with the county name.

PDF



June 25,2019 — Delinquent tax payer report — DelinquentTaxPayer.rpt

11:03 AM — Page 1 of 163

Carroll
2018 Pay 2019

Property number	Lender Id	Name	Location street address	Prior years Tax Fee	Prior years Penalty Cost	Spring Tax	Spring Penalty	Fall Tax Fee	Fall Penalty Cost	Total Tax Fee	Total Penalty Cost
001 - ADAMS TWP.											
08-02-00-000-006.000-001		Quaglio, Leonard	1441 W Towpath Rd, BURNETTSVILLE IN 47926	188.54	18.86	203.56	39.21	203.56	0.00	595.66	58.07
				0.00	0.00			0.00	0.00	0.00	0.00
08-02-00-000-009.000-001		Shelton, Jason G.	N 300 W , BURNETTSVILLE IN 47926	0.00	0.00	0.00	0.79	15.68	0.00	15.68	0.79
				0.00	0.00			0.00	0.00	0.00	0.00
08-02-00-000-028.000-001		Goodner, Richard M	W Towpath Rd, Delphi IN 46923	16.94	5.10	6.06	2.31	0.00	0.00	23.00	7.41
				0.00	0.00			0.00	0.00	0.00	0.00
08-02-00-000-036.000-001		Ratcliff, Glen Alan	N 350 W , Burnettsville IN 47926	423.28	62.64	149.76	57.31	149.76	0.00	722.80	119.95
				0.00	0.00			130.00	0.00	130.00	0.00
08-02-00-000-037.000-001		Shelton, Jason G.	N 300 W , BURNETTSVILLE IN 47926	0.00	0.00	0.00	0.36	0.00	0.00	0.00	0.36
				0.00	0.00			0.00	0.00	0.00	0.00
08-02-04-000-015.000-001		Welker, Maryjane LE ETAL	12105 N 50 W , Burnettsville IN 47926	0.00	20.93	224.37	44.74	224.37	0.00	448.74	65.67
				0.00	0.00			0.00	0.00	0.00	0.00
08-02-05-000-001.000-001		Moon, Richard D & Diana F	W 1200 N , Burnettsville IN 47926	0.00	0.00	830.75	83.08	830.75	0.00	1,661.50	83.08
				0.00	0.00			0.00	0.00	0.00	0.00
08-02-05-000-002.000-001		Moon, Richard D & Diana F	W 1300 N , Burnettsville IN 47926	0.00	0.00	88.03	8.80	88.03	0.00	176.06	8.80
				0.00	0.00			0.00	0.00	0.00	0.00
08-02-05-000-003.000-001		Moon, Richard & Diana	W 1300 N , Burnettsville IN 47926	0.00	0.00	214.01	21.40	214.01	0.00	428.02	21.40
				0.00	0.00			0.00	0.00	0.00	0.00
08-02-05-000-008.000-001		Moon, Richard D & Diana F	1736 W 1250 N , Burnettsville IN 47926	0.00	0.00	535.54	53.55	535.54	0.00	1,071.08	53.55
				0.00	0.00			0.00	0.00	0.00	0.00
08-02-05-000-010.000-001		Moon, Richard D & Diana F	W 1250 N , Burnettsville IN 47926	0.00	0.00	704.26	70.43	704.26	0.00	1,408.52	70.43
				0.00	0.00			0.00	0.00	0.00	0.00
08-02-06-000-021.000-001		Stoller, John D	12502 N 250 W , BURNETTSVILLE IN 47926	0.00	0.00	146.08	14.61	146.08	0.00	292.16	14.61
				0.00	0.00			0.00	0.00	0.00	0.00
08-02-06-000-022.000-001		Stoller, John D	250 W , BURNETTSVILLE IN 47926	0.00	0.00	17.20	1.72	0.00	0.00	17.20	1.72
				0.00	0.00			0.00	0.00	0.00	0.00
08-02-06-000-025.000-001		Coble, Randy & Brenda S.	2153 W 1200 N , Burnettsville IN 47926	0.00	0.00	427.52	42.75	427.52	0.00	855.04	42.75
				0.00	0.00			0.00	0.00	0.00	0.00

Python Code (20190712_ExtractPDF.py)

```
# Read and dump PDF file data to text file

# Accumatch

# by Michael Keller

# June 28, 2019

# requires pdfplumber, re


# libraries

import pdfplumber

import re


# Booleans

removeCommas = True

makeCSV = True

countyFound = False

Property_Found = False

propertyFound = False


# Variables

outputType = '.csv'

outputFile = ''

filename = ''

count = 0

outString = []

tempString = ''

lineLength = 20

chars = set(' 0123456789.')

spaces = '\s{2,}'

space = ' '

nospace = ''

comma = ','

period = '.'

_return = '\r'

regProperty = '\d{2}[-]\d{2}[-]\d{2}[-]\d{3}[-]\d{3}[.]\d{3}[-]\d{3}'

colHeadings = 'Property Number,Prior Years Tax,Prior Years Penalty,Spring Tax,Spring Penalty,Fall
Tax,Fall Penalty,Total Tax,Total Penalty,Prior Years Fee,Prior Years Cost,Fall Tax Fee,Fall
Penalty Cost,Total Tax Fee,Total Penalty Cost'
```

```
# Messages
messageRun = 'Reading \'input.pdf\' file . . .'
messageSaveBeg = 'Saving \''
messageSaveEnd = '\' file . . .'
messageEOL = '. . . end of line'

# PDF File Open
pdf = pdfplumber.open("input.pdf")

print(messageRun)

for page in pdf.pages:
    p1 = pdf.pages[count]
    count = count + 1
    pltext = p1.extract_text()
    text = pltext.splitlines()

    for item in text:
        # Search for Property number in line read from PDF
        Property_Found = re.search(regProperty, item)
        # First line of good data in PDF
        if(Property_Found is not None):
            propertyFound = True
            Property = item[Property_Found.start():Property_Found.end()]
            # Grab the rest of the line using the Property found variable
            restOfLine = item[Property_Found.end():len(item)].strip()
            # Take out commas in amounts and other fields
            restOfLine = restOfLine.replace(comma,nospace)
            # Remove double spaces in rest of line
            restOfLine_mod = re.sub(spaces, space, restOfLine)
            # Split from the left nine ways
            carp = restOfLine_mod.rsplit(' ',9)
            # Add only the last nine items or amount found from the split
            tempString = str(Property) + comma + str(carp[2]) + comma + str(carp[3]) + comma +
str(carp[4]) + comma + str(carp[5]) + comma + str(carp[6]) + comma + str(carp[7]) + comma +
str(carp[8]) + comma + str(carp[9])
```

```
        elif((propertyFound) and (len(item) > lineLength) and (set(item).issubset(chars))):
            propertyFound = False
            # Add blank to make lines from different files similar
            carp1 = ' ' + item
            #print(carp1)
            # Take out commas in amounts and other fields
            carp2 = carp1.replace(comma,nospace)
            # Remove double spaces in rest of line
            carp3 = re.sub(spaces, space, carp2)
            # Split from the left five ways
            carp4 = carp3.rsplit(' ',6)
            outString.append(tempString + comma + str(carp4[1]) + comma + str(carp4[2]) + comma +
str(carp4[3]) + comma + str(carp4[4]) + comma + str(carp4[5]) + comma + str(carp4[6]))
        else:
            # Numbers change on the Howard file at page 277
            if(countyFound == False):
                # Do not run this if statement again i.e. change boolean to True
                countyFound = True
                # Read first line in PDF and create filename based on the text found
                filename = text[3].strip()
                # Create filename based on county name found in PDF
                outputFile = filename + outputType


#PDF File Close
pdf.close()


print(messageSaveBeg + outputFile + messageSaveEnd)


# Create and open output file
file = open(outputFile,'w')


# Write out lines from array
file.write(colHeadings + _return)


# Now dump string array into file
for taxLine in outString:
    if (makeCSV):
```

```
        taxLine = taxLine.replace(space,comma)
    file.write(taxLine + _return)

print(messageEOL)

file.close()

# EOF
```

Python Output

```
 RESTART: G:\Documents\Code\Python\code\20190712_ExtractPDF\20190712_ExtractPDF.py
Reading 'input.pdf' file . . .
Saving 'Carroll.csv' file . . .
. . . end of line
>>>
 RESTART: G:\Documents\Code\Python\code\20190712_ExtractPDF\20190712_ExtractPDF.py
Reading 'input.pdf' file . . .
Saving 'Howard.csv' file . . .
. . . end of line
>>>
```

CSV Output

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Property Number	Prior Years Tax	Prior Years Penalty	Spring Tax	Spring Penalty	Fall Tax	Fall Penalty	Total Tax	Total Penalty	Prior Years Fee	Prior Years Cost	Fall Tax Fee	Fall Penalty Cost	Total Tax Fee	Total Penalty Cost
2	08-02-00-000-006.000-001	188.54	18.86	203.56	39.21	203.56	0	595.66	58.07	0	0	0	0	0	0
3	08-02-00-000-009.000-001	0	0	0	0.79	15.68	0	15.68	0.79	0	0	0	0	0	0
4	08-02-00-000-028.000-001	16.94	5.1	6.06	2.31	0	0	23	7.41	0	0	0	0	0	0
5	08-02-00-000-036.000-001	423.28	62.64	149.76	57.31	149.76	0	722.8	119.95	0	0	130	0	130	0
6	08-02-00-000-037.000-001	0	0	0	0.36	0	0	0	0.36	0	0	0	0	0	0
7	08-02-04-000-015.000-001	0	20.93	224.37	44.74	224.37	0	448.74	65.67	0	0	0	0	0	0
8	08-02-05-000-001.000-001	0	0	830.75	83.08	830.75	0	1661.5	83.08	0	0	0	0	0	0
9	08-02-05-000-002.000-001	0	0	88.03	8.8	88.03	0	176.06	8.8	0	0	0	0	0	0
10	08-02-05-000-003.000-001	0	0	214.01	21.4	214.01	0	428.02	21.4	0	0	0	0	0	0
11	08-02-05-000-008.000-001	0	0	535.54	53.55	535.54	0	1071.08	53.55	0	0	0	0	0	0
12	08-02-05-000-010.000-001	0	0	704.26	70.43	704.26	0	1408.52	70.43	0	0	0	0	0	0
13	08-02-06-000-021.000-001	0	0	146.08	14.61	146.08	0	292.16	14.61	0	0	0	0	0	0
14	08-02-06-000-022.000-001	0	0	17.2	1.72	0	0	17.2	1.72	0	0	0	0	0	0
15	08-02-06-000-025.000-001	0	0	427.52	42.75	427.52	0	855.04	42.75	0	0	0	0	0	0
16	08-02-06-000-026.000-001	158.99	15.9	170	17	170	0	498.99	32.9	0	0	0	0	0	0
17	08-02-07-000-006.000-001	0	0	185.02	18.5	185.02	0	370.04	18.5	0	0	0	0	0	0
18	08-02-07-000-012.000-001	1659.69	394.82	520.69	218.04	520.69	0	2701.07	612.86	0	0	130	0	130	0
19	08-02-18-000-007.000-001	0	0	126.99	12.7	126.99	0	253.98	12.7	0	0	0	0	0	0
20	08-02-18-000-062.000-001	39.56	15.06	14.16	5.38	0	0	53.72	20.44	75	30	130	0	205	30
21	08-02-18-000-063.000-001	217.17	36.84	76.87	29.41	76.87	0	370.91	66.25	0	0	130	0	130	0
22	08-02-18-000-115.000-001	5.68	0.57	6.06	1.18	0	0	11.74	1.75	0	0	0	0	0	0
23	08-03-00-000-142.000-001	0	0	0	0.71	0	0	0	0.71	0	0	0	0	0	0
24	08-03-00-000-145.000-001	0	0	14.16	1.42	14.16	0	28.32	1.42	0	0	0	0	0	0
25	08-03-00-000-160.000-001	789.25	78.92	839.86	162.92	839.86	0	2468.97	241.84	0	0	0	0	0	0
26	08-03-01-000-010.000-001	0	0	230.71	23.07	230.71	0	461.42	23.07	0	0	0	0	0	0
27	08-03-01-000-014.000-001	0	0	352.64	35.26	352.64	0	705.28	35.26	0	0	0	0	0	0
28	08-03-03-000-013.000-001	0	0	406.78	40.68	406.78	0	813.56	40.68	0	0	0	0	0	0

Michael Keller
Accumatch
20190712_ExtractPDF
July 17, 2019