

Delinquent List

Notes: This python scripts scrapes PDFs creates a CSV output file with the following items: **Account ID** and **Balance**. The first page of the PDF contains not tax data and is skipped The filename is static and not unique.

PDF

Delinquent Tax Listing

Account Id	Owner	Balance
0493390	3 AQUILA L L C	\$1,725.01
0088788	3G2B INVESTMENTS L L C	\$5,011.85
0205366	4 STAR KIDS RANCH TRUST GODFREY JEVF TRUSTEE	\$66.80
0493842	33 ALL IN ENTERPRISES L L C	\$5,684.96
0225950	100% LOCATION L L C UND 8% INT PENSCO TRUST COMPANY/CUSTODIAN 92% INT BOCEK JANET RUTH IRA	\$61.17
0151412	508 UTAH ACRES L C	\$509.41
0218872	2015 MOORE FAMILY TRUST MOORE JOE A TRUSTEE MOORE SUSAN R TRUSTEE	\$334.84
0271129	1256454 ALBERTA L L C	\$7,034.00
0311784	2094066 ONTARIO INC	\$33.23
0306867	AASCE SCOTT	\$61.17
0306701	AASCE SCOTT	\$61.17
0359270	ABASTA BERNICE R REVOCABLE LIVING TRUST ABASTA BERNICE R TRUSTEE	\$0.00
0184256	ABATE DAVIDE A	\$33.23
0238490	ABBOTT EDWARD V	\$66.47
0177912	ABC BRICK SPECIALIST ALGER STEVE	\$10.00
0494302	ABLAHANI JOHN ABRAHAM	\$20.54
0491139	ABRAMS BRAD/RACHEL J/T ABRAMS RACHEL J/T	\$312.79
0410818	ABRAMS WILLIAM M/JULIE R J/T ABRAMS JULIE R J/T	\$20.39
0165297	ABUJAYYAB MOHAMMED A J/T ZIMMERMAN ASHLEIGH V J/T	\$255.62
0106788	ACE JACKSON CORPORATION	\$107.22
0277555	ACE STORAGE RENTALS L L C	\$13,266.35
0308350	ACE STORAGE RENTALS L L C	\$15,185.47
0264850	ACEVEDO GILBERTO DE JOEL NOVA	\$103.05
0092525	ACEVEDO JOSE M	\$86.76
0468303	ACH ACQUISITIONS L L C	\$1,064.37

Python Code (20190716_DelinquentList.py)

```
# Read and dump PDF file data to text file for Lender Tax List
# Accumatch
# by Michael Keller
# July 16, 2019
# requires pdfplumber, re, datetime

# Libraries
import pdfplumber
import re
from datetime import datetime

# Start Timer
start = datetime.now()

# Booleans
Account_Found = False

# Variables
account = ''
balance = ''
dollarSign = '$'
outputType = '.csv'
outputFile = 'Delinquent-Tax-Listing'
filename = ''
count = 0
outString = []
nospace = ''
comma = ','
return = '\r'
regAccount = '^[0-9]+'
regDollar = '$'
colHeadings = 'Account Id,Balance'

# Messages
messageRun = 'Reading \'input.pdf\' file . . .'
```

```
messageSaveBeg = 'Saving \''
messageSaveEnd = '\' file . . .'
messageEOL = '. . . end of line'

# PDF File Open
pdf = pdfplumber.open("Delinquent-Tax-Listing.pdf")

print(messageRun)

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

## Break for Testing
## if(count == 195):
##     break

    for item in text:
        # Search for Account number in line read from PDF
        Account_Found = re.search(regAccount, item)

        if(Account_Found is not None):
            account = item[0:7].strip()
            carp = item.rsplit(' ',2)
            carp2 = carp[2].replace(comma,nospace)
            carp2 = carp2.replace(dollarSign,nospace)
            balance = carp2.strip()
            outString.append(account + comma + balance)

# PDF File Close
pdf.close()

filename = outputFile + outputFile
print(messageSaveBeg + filename + messageSaveEnd)
```

```
# Create and open output file
try:
    file = open(filename, 'w')
    # Write out lines from array
    file.write(colHeadings + _return)
    # Now dump string array into file
    for taxLine in outString:
        file.write(taxLine + _return)
    file.close()
except PermissionError:
    print('ERROR: Unable able to open output file for writing. Is the file currently open?')

#end timer
print('Runtime: ' + str(datetime.now() - start))

print(messageEOL)

# EOF
```

CSV Output

	A	B	
1	Account Id	Balance	
2	493390	1725.01	
3	88788	5011.85	
4	205366	66.8	
5	493842	5684.96	
6	225950	61.17	
7	151412	509.41	
8	218872	334.84	
9	271129	7034	
10	311784	33.23	
11	306867	61.17	
12	306701	61.17	
13	359270	0	
14	184256	33.23	
15	238490	66.47	
16	177912	10	
17	494302	20.54	
18	491139	312.79	
19	410818	20.39	
20	165297	255.62	
21	106788	107.22	
22	277555	13266.35	
23	308350	15185.47	
24	264850	103.05	
25	92525	86.76	
26	468303	1064.37	
27	193273	33.23	
28	296076	102.11	

Michael Keller
Accumatch
20190716_DelinquentList
July 17, 2019