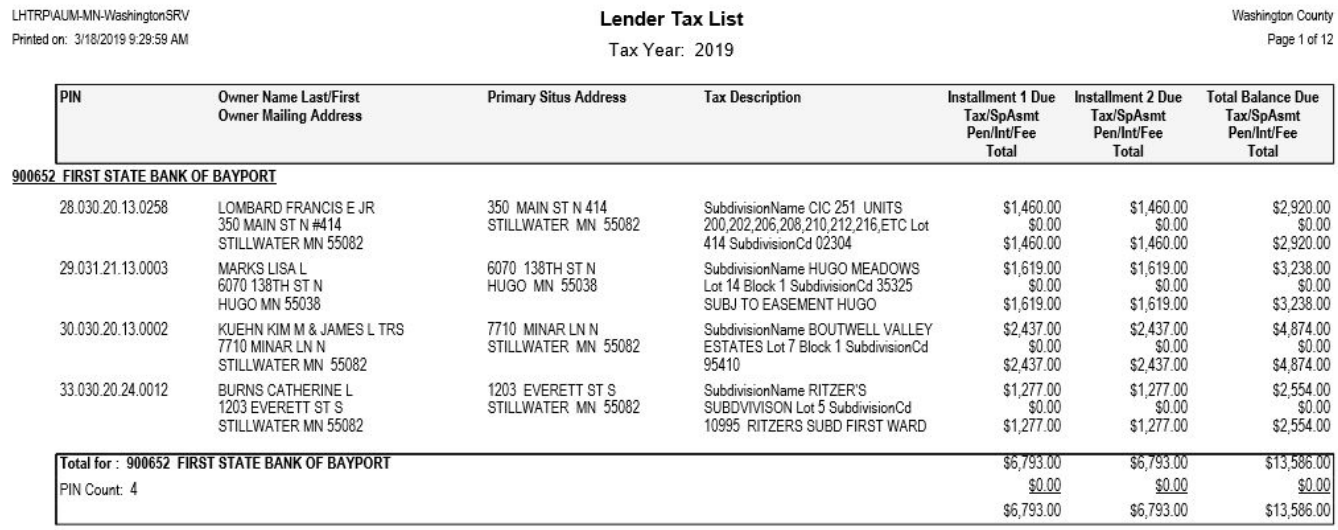
Lender Tax List

Notes: This python scripts scrapes PDFs creates a CSV output file with the following items: **PIN, Installment 1 Due, Installment 2 Due, Total Balance Due, Installment 1 Tax, Installment 2 Tax Total SP Amount, Install 1 Fee, Install 2 Fee, Total Fee**. The filename is static and not unique.

PDF



LHTRP\AUM-MN-WashingtonSRV
Printed on: 3/18/2019 9:29:59 AM

Lender Tax List
Tax Year: 2019

Washington County
Page 1 of 12

PIN	Owner Name Last/First Owner Mailing Address	Primary Situs Address	Tax Description	Installment 1 Due Tax/SpAsmt Pen/Int/Fee Total	Installment 2 Due Tax/SpAsmt Pen/Int/Fee Total	Total Balance Due Tax/SpAsmt Pen/Int/Fee Total
900652 FIRST STATE BANK OF BAYPORT						
28.030.20.13.0258	LOMBARD FRANCIS E JR 350 MAIN ST N #414 STILLWATER MN 55082	350 MAIN ST N 414 STILLWATER MN 55082	SubdivisionName CIC 251 UNITS 200,202,206,208,210,212,216,ETC Lot 414 SubdivisionCd 02304	$1,460.00 $0.00 $1,460.00	$1,460.00 $0.00 $1,460.00	$2,920.00 $0.00 $2,920.00
29.031.21.13.0003	MARKS LISA L 6070 138TH ST N HUGO MN 55038	6070 138TH ST N HUGO MN 55038	SubdivisionName HUGO MEADOWS Lot 14 Block 1 SubdivisionCd 35325 SUBJ TO EASEMENT HUGO	$1,619.00 $0.00 $1,619.00	$1,619.00 $0.00 $1,619.00	$3,238.00 $0.00 $3,238.00
30.030.20.13.0002	KUEHN KIM M & JAMES L TRS 7710 MINAR LN N STILLWATER MN 55082	7710 MINAR LN N STILLWATER MN 55082	SubdivisionName BOUTWELL VALLEY ESTATES Lot 7 Block 1 SubdivisionCd 95410	$2,437.00 $0.00 $2,437.00	$2,437.00 $0.00 $2,437.00	$4,874.00 $0.00 $4,874.00
33.030.20.24.0012	BURNS CATHERINE L 1203 EVERETT ST S STILLWATER MN 55082	1203 EVERETT ST S STILLWATER MN 55082	SubdivisionName RITZER'S SUBDIVIVISON Lot 5 SubdivisionCd 10995 RITZERS SUBD FIRST WARD	$1,277.00 $0.00 $1,277.00	$1,277.00 $0.00 $1,277.00	$2,554.00 $0.00 $2,554.00
Total for : 900652 FIRST STATE BANK OF BAYPORT				$6,793.00	$6,793.00	$13,586.00
PIN Count: 4				$0.00	$0.00	$0.00
				$6,793.00	$6,793.00	$13,586.00

Python Code (20190716_LenderTaxList.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

PIN_Found = False

Case_Found = False

Dollar_Found = False

pinFound = False


# Variables

pin = ''

install1Due = ''

install2Due = ''

totalDue = ''

install1Tax = ''

install2Tax = ''

totalSpAmt = ''

install1Fee = ''

install2Fee = ''

totalFee = ''

dollarSign = '$'

outputType = '.csv'

outputFile = 'LenderTaxList'

filename = ''
```

```
count = 0

outString = []

nospace = ''

comma = ','

_return = '\r'

regPIN = '\d{2}[.]\d{3}[.]\d{2}[.]\d{2}[.]\d{4}'

regDollar = '[$]'

colHeadings = 'PIN,Installment 1 Due,Installment 2 Due,Total Balance Due,Installment 1
Tax,Installment 2 Tax,Total SP Amount,Install 1 Fee,Install 2 Fee,Total Fee'


# 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()


##    Break for Testing

##    if(count == 2):

##        break


    for item in text:

        # Search for Parcel number in line read from PDF

        PIN_Found = re.search(regPIN, item)

        # Search for any kind of dollar agmount

        Dollar_Found = re.search(regDollar, item)
```

```
        if(PIN_Found is not None):
            pinFound = True
            pin = item[PIN_Found.start():PIN_Found.end()]
            restOfLine = item[PIN_Found.end():len(item)].strip()
            restOfLine = restOfLine.replace(comma,nospace)
            restOfLine = restOfLine.replace(dollarSign,nospace)
            carp = restOfLine.rsplit(' ',3)
            install1Due = carp[1].strip()
            install2Due = carp[2].strip()
            totalDue = carp[3].strip()
            tempString = pin + comma + install1Due + comma + install2Due + comma + totalDue +
comma
            lineCount = 0
        elif((pinFound) and (Dollar_Found is not None)):
            lineCount = lineCount + 1
            if(lineCount == 1):
                # get second line of data
                restOfLine = item.replace(comma,nospace)
                restOfLine = restOfLine.replace(dollarSign,nospace)
                carp = restOfLine.rsplit(' ',3)
                install1Tax = carp[1].strip()
                install2Tax = carp[2].strip()
                totalSpAmt = carp[3].strip()
                tempString = tempString + str(install1Tax) + comma + str(install2Tax) + comma +
str(totalSpAmt) + comma
            else:
                # get third line of data
                restOfLine = item.replace(comma,nospace)
                restOfLine = restOfLine.replace(dollarSign,nospace)
                carp = restOfLine.rsplit(' ',3)
                install1Fee = carp[1].strip()
                install2Fee = carp[2].strip()
                totalFee = carp[3].strip()
                outString.append(tempString + str(install1Fee) + comma + str(install2Fee) + comma
+ str(totalFee))
                pinFound = False


# PDF File Close
```

```
pdf.close()


filename = outputFile + outputType
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	C	D	E	F	G	H	I	J
1	PIN	Installment 1 Due	Installment 2 Due	Total Balance Due	Installment 1 Tax	Installment 2 Tax	Total SP Amount	Install 1 Fee	Install 2 Fee	Total Fee
2	28.030.20.13.0258	1460	1460	2920	0	0	0	1460	1460	2920
3	29.031.21.13.0003	1619	1619	3238	0	0	0	1619	1619	3238
4	30.030.20.13.0002	2437	2437	4874	0	0	0	2437	2437	4874
5	33.030.20.24.0012	1277	1277	2554	0	0	0	1277	1277	2554
6	02.028.20.12.0038	1156	1156	2312	0	0	0	1156	1156	2312
7	02.028.20.34.0021	1707	1707	3414	0	0	0	1707	1707	3414
8	02.029.20.31.0003	430	430	860	0	0	0	430	430	860
9	02.029.20.31.0005	134	134	268	0	0	0	134	134	268
10	02.032.21.22.0006	2075	2075	4150	0	0	0	2075	2075	4150
11	03.029.20.24.0014	1339	1339	2678	0	0	0	1339	1339	2678
12	03.029.20.44.0024	863	863	1726	0	0	0	863	863	1726
13	03.029.20.44.0030	608	608	1216	0	0	0	608	608	1216
14	03.029.20.44.0074	674	674	1348	0	0	0	674	674	1348
15	04.029.20.11.0045	393	393	786	0	0	0	393	393	786
16	04.029.20.11.0047	393	393	786	0	0	0	393	393	786
17	04.029.20.11.0093	872	872	1744	0	0	0	872	872	1744
18	04.029.20.11.0131	1131	1131	2262	0	0	0	1131	1131	2262
19	04.029.20.11.0144	1035	1035	2070	0	0	0	1035	1035	2070
20	04.029.20.13.0043	1516	1516	3032	0	0	0	1516	1516	3032
21	04.029.20.14.0069	1945	1945	3890	0	0	0	1945	1945	3890
22	04.029.20.21.0078	1050	1050	2100	0	0	0	1050	1050	2100
23	04.029.20.23.0011	1202	1202	2404	0	0	0	1202	1202	2404
24	04.029.20.32.0055	1304	1304	2608	0	0	0	1304	1304	2608
25	04.029.20.32.0078	1792	1792	3584	0	0	0	1792	1792	3584
26	05.029.20.11.0009	533	533	1066	0	0	0	533	533	1066
27	06.027.21.41.0045	2773	2773	5546	0	0	0	2773	2773	5546
28	07.028.20.12.0008	1592	1592	3184	0	0	0	1592	1592	3184

Michael Keller
Accumatch
20190716_LenderTaxList
July 17, 2019