

## Forfeited Land Sale List (Properties Sold)

**Notes:** This python scripts scrapes PDFs creates a CSV output file with the following items:

**Book/Plat/Parcel, Case #, Land, Building, Court Cost, Taxes, Total, Purchase Amt.** The filename is static and not unique.

PDF



Dusty Rhodes  
Hamilton County Auditor

### Results of the 2019 Forfeited Land Sale

--- Properties Sold ---

<u>Item:</u>	<u>Book/Plat/Parcel:</u>	<u>Owner</u>	<u>Property Description:</u>	<u>Case No:</u>	<u>Land/ Building:</u>	<u>Court Cost/ Taxes/ Total:</u>	<u>Purchased By:</u>	<u>Amount:</u>
1	016-0001-0086-00	BOWLING BILLY P	4500 EASTERN AV 50 X 98.50 IR LOT 302-303 UNDERCLIFF 1ST SUB PARS 86-87 CONS	A1607175	20,790 20,130	\$1,166.99 \$11,815.55 \$12,982.54	THOMPSON ERICK 2144 SAINT LEO PL CINCINNATI, OH 45211	800.00
2	024-0003-0126-00	ACTIVE HOMESLLC	4456 EASTERN AVE 25 X 91.50 IRR LOT 259 UNDERCLIFFS 1ST	A1605549	5,870 0	\$711.10 \$10,398.75 \$11,109.85	TUSCULUM RIDGE LLC 4242 AIRPORT RD SUITE 108 CINCINNATI, OH 45226	500.00
3	024-0003-0167-00	MGDK INVESTMENTS LLC	4415 EASTERN AVE 25 X 229.84 IRR LOT 551 UNDERCLIFFS 2ND	A1606430	8,370 0	\$779.40 \$14,773.18 \$15,552.58	TUSCULUM RIDGE LLC 4242 AIRPORT RD SUITE 108 CINCINNATI, OH 45226	500.00
4	024-0003-0194-00	TAYLOR KENNETH F TR	EASTERN AVE 66.67 X 93.50 PTS LOTS 521-522-523 UNDER- CLIFFS 2ND SUB	A1701072	10,890 21,560	\$912.09 \$9,423.34 \$10,335.43	TUSCULUM RIDGE LLC 4242 AIRPORT RD SUITE 108 CINCINNATI, OH 45226	600.00
5	027-0006-0101-00	SMITH RICHARD	247 MC CULLOUGH ST 25 X 189 PT LOT 4 MC MAHON EST SUB	A1702962	10,200 0	\$996.40 \$12,756.64 \$13,753.04	VANCY INVESTMENTS LLC 7360 RIVERBY DR CINCINNATI, OH 45255	600.00
6	034-0002-0099-00	JONES CORA WAGNER ET AL	HIGH ST 168.50 X 242 LOT 78 WHITE & ROSS SUB	A1601293	10,180 0	\$1,759.80 \$4,960.58 \$6,720.38	THOMPSON ERICK 2144 SAINT LEO PL CINCINNATI, OH 45211	500.00
7	037-0001-0205-00	DESILVA BERNARD	4207 WHETSEL AVE 40 X 150 LOT 6 C A HEILKERS OF LOTS 139 140 SUB	A1704521	17,380 0	\$1,046.48 \$24,493.13 \$25,539.61	AP RIMTEX LLC 6454 TIMBERS CT MASON, OH 45040	2,500.00

## Python Code (20190716\_ForLandSold.py)

```
# Read and dump PDF file data to text file for properties sold
# 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
Parcel_Found = False
Case_Found = False
Dollar_Found = False
parcelFound = False

# Variables
parcel = ''
case = ''
land = ''
court_cost = ''
purchase_amt = ''
outputType = '.csv'
outputFile = 'PropertiesNotSold'
filename = ''
count = 0
outString = []
tempString = ''
nospace = ''
comma = ','
dollarSign = '$'
```

```

_return = '\r'
regParcel = '\d{3}[-]\d{4}[-]\d{4}[-]\d{2}'
regCase = '[a-zA-Z]\d{7}'
regNumbers = '^([0-9.]+)$'
regDollar = '[$]'
colHeadings = 'Book/Plat/Parcel,Case #,Land,Building,Court Cost,Taxes,Total,Purchase Amt'

# 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 Parcel number in line read from PDF
        Parcel_Found = re.search(regParcel, item)

        # Search for any kind of dollar agmount
        Dollar_Found = re.search(regDollar, item)

        # First line of good data in PDF
        if(Parcel_Found is not None):
            parcelFound = True
            parcel = item[Parcel_Found.start():Parcel_Found.end()]

            Case_Found = re.search(regCase, item)
            case = item[Case_Found.start():Case_Found.end()]

```

```

restOfLine = item[Case_Found.end():len(item)].strip()
restOfLine = restOfLine.replace(commma,nospace)
carp = restOfLine.split(' ',2)
land = carp[0].strip()
court_cost = carp[1].strip()
court_cost = court_cost.replace(dollarSign,nospace)
restOfLine = carp[2].strip()
carp = restOfLine.rsplit(' ',1)
carp2 = carp[1].strip()
if(re.match(regNumbers,carp2)):
    purchase_amt = carp2
else:
    purchase_amt = '0.0'
tempString = parcel + commma + case + commma + land + commma
lineCount = 0
elif((parcelFound) and (Dollar_Found is not None)):
    lineCount = lineCount + 1
    if(lineCount == 1):
        # get remaining amounts from second line
        findDollar = item.find(dollarSign)
        begOfLine = item[0:findDollar]
        restOfLine = item[findDollar+1:len(item)].strip()
        carp = begOfLine.rsplit(' ',2)
        carp1 = restOfLine.split(' ',1)
        taxes = carp1[0].replace(commma,nospace)
        building = carp[1].strip()
        building = building.replace(commma,nospace)
        tempString = tempString + str(building) + commma + str(court_cost) + commma +
str(taxes)
    else:
        # get total
        carp = item[Dollar_Found.start()+1:len(item)] + ' '
        carp = carp.split(' ',2)
        total = carp[0].replace(commma,nospace).strip()
        outString.append(tempString + commma + total + commma + purchase_amt)
        parcelFound = False

```

```
# 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	C	D	E	F	G	H
1	Book/Plat/Parcel	Case #	Land	Building	Court Cost	Taxes	Total	Purchase Amt
2	016-0001-0086-00	A1607175	20790	20130	1166.99	11815.55	12982.54	800
3	024-0003-0126-00	A1605549	5870	0	711.1	10398.75	11109.85	500
4	024-0003-0167-00	A1606430	8370	0	779.4	14773.18	15552.58	500
5	024-0003-0194-00	A1701072	10890	21560	912.09	9423.34	10335.43	600
6	027-0006-0101-00	A1702962	10200	0	996.4	12756.64	13753.04	600
7	034-0002-0099-00	A1601293	10180	0	1759.8	4960.58	6720.38	500
8	037-0001-0205-00	A1704521	17380	0	1046.48	24493.13	25539.61	2500
9	054-0004-0094-00	A1700881	8500	30710	0.01	8267.05	8267.06	1600
10	065-0002-0012-00	A1704189	15720	73170	752.8	32917.1	33669.9	600
11	090-0002-0023-00	A1703880	12360	0	904.29	27502.4	28406.69	400
12	096-0002-0024-00	A1600946	1440	33150	913.3	7615.32	8528.62	400
13	098-0004-0091-00	A1702020	15300	0	919.79	4159.03	5078.82	400
14	098-0004-0172-00	A1702122	15000	0	912.29	5000.59	5912.88	100
15	098-0005-0074-00	A1705436	13650	0	873.89	23181.34	24055.23	600
16	098-0005-0099-00	A1603637	15400	33610	682	12887.46	13569.46	300
17	099-0002-0066-00	A1605983	23320	0	925.29	27020.46	27945.75	50
18	107-0010-0106-00	A1704520	11830	0	867.89	42067.8	42935.69	700
19	107-0011-0040-00	A1705178	10810	21480	923.99	16846.39	17770.38	11050
20	107-0011-0074-00	A1604497	10810	0	731.3	19448.29	20179.59	3000
21	108-0001-0019-00	A1606048	9280	0	1040.08	3997.86	5037.94	475
22	111-0005-0062-00	A1604312	8160	0	0.01	31026.62	31026.63	899
23	111-0005-0084-00	A1604312	30	0	852.48	37.54	890.02	1
24	115-0001-0281-00	A1603042	16800	57120	801.1	21821.66	22622.76	8050
25	116-0002-0173-00	A1600958	33660	0	706.4	17044.78	17751.18	8000
26	117-0008-0105-00	A1606943	21390	0	728	6422.79	7150.79	5000
27	117-0012-0142-00	A1700427	2230	0	1063.69	3442.62	4506.31	4581.31
28	120-0003-0070-00	A1705260	32790	0	1019.58	33826.41	34845.99	9500

Michael Keller  
Accumatch  
20190716\_ForflandSold  
July 17, 2019