

How to Find Total Length of Service for Volunteers in an OU

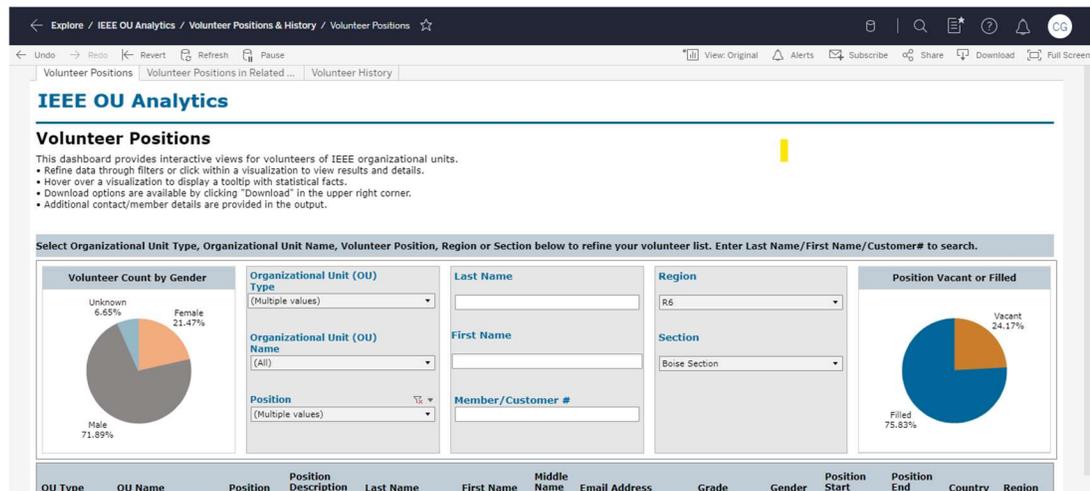
This document describes how to get a report of the current and past volunteers within an OU, the total length of time they've served in each position, and the time they've currently served in the position.

Sometimes it is useful to determine how long volunteers have served in a position in a section or chapter. For example, it may be helpful to identify more seasoned volunteers for their experience and as potential mentors, and newer volunteers as potential mentees.

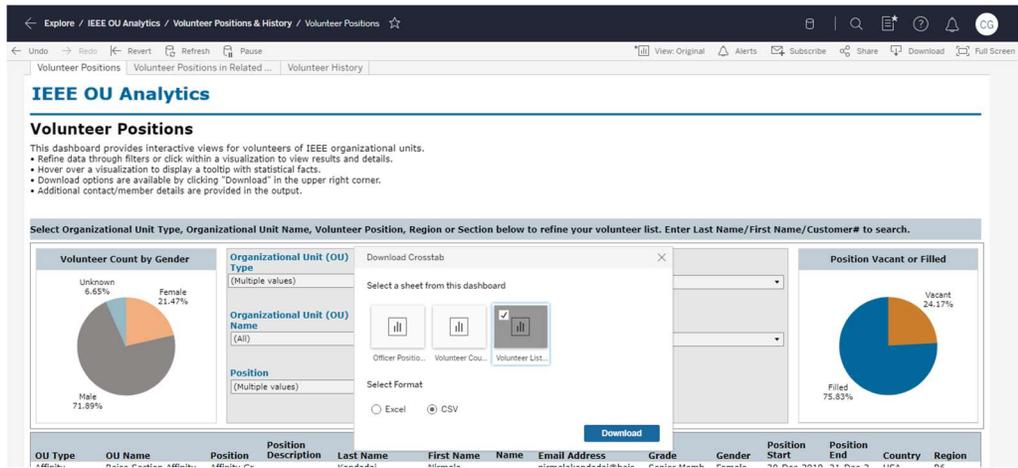
This report can also be helpful to comply with MGA policy. The MGA Operations Manual states "All officers shall not serve in any one position, in any single organizational unit, more than six years in total." This policy refers to section and chapter ExCom officers (chair, vice chair, treasurer, secretary.) Exceptions can be approved by the parent OU, for example by the section chair for chapter and affinity group officers and by the Region Director for section officers.

The volunteer data in OU Analytics has the information needed to find out the total length of service of a volunteer in a position. However, if the volunteer has served in the position over multiple time periods, data must be combined from the "Volunteer Positions" tab and the "Volunteer History" tab of the "Volunteer Positions & History" workbook. This is cumbersome to do manually, but a Python script is available to do this. This script is attached below.

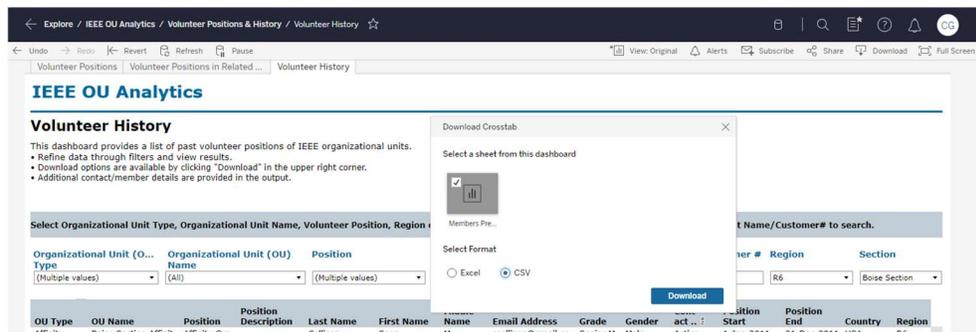
1. In the OU Analytics "Volunteer Positions" workbook, select the OU Type and Positions of interest, and the Region and/or Section of interest. In this example, the chair, vice chair, secretary, treasurer, and secretary/treasurer positions are selected for all Affinity, Chapter, and Section OU Types in the Boise section.



2. Download the "Volunteer List by OU" CSV file. For this example, I'll rename the downloaded file to "Boise current officers.csv"



3. Apply the same filters in the Volunteer History tab and download the CSV file. For this example I'll rename the downloaded file to "Boise previous officers.csv"



4. If you don't already have Python installed on your PC, install it. For this example I'm using Python 3.8.2 which I downloaded from <https://python.org/downloads>.
5. Download the Python script `ieeeyearsInPosition.py.txt` and rename it to `ieeeyearsInPosition.py`. For this example I placed it in the same folder as the CSV files I downloaded earlier. The source code is also at the bottom of this document.
6. Open a command prompt, change the current directory to the folder containing the two downloaded CSV files, and run the Python script. The first command line argument after the script name is the name of the current volunteer CSV file. The second command line argument is the name of the volunteer history CSV file. Redirect the output to a file. In this example I redirect to "Boise officers years in position.csv"

```

C:\Users\chris\Downloads>python ieeeyearsInPosition.py "Boise current officers.csv" "Boise previous officers.csv" > "Boise officers years in position.csv"
C:\Users\chris\Downloads>

```

7. Using either Excel or Google Sheets, create a blank worksheet and import the csv file that was created by the Python script. In the example below I use Google Sheets. I've set up filters to only show current officers and I sorted the Years Served column in descending order. I also blurred the names and member numbers below though that data is available to any volunteer that has access to OU Analytics.

The "Years Served" column shows the total number of years the volunteer has served in the position, while the "Current Term" column shows how long they've been currently serving in the position. You can see in Row#14 that that volunteer has served 3.2 years in the position, but there was a break in service; he has served 0.7 years in his current term of service.

Name: Filter 1 Range: A1:Z181

	A	B	C	D	E	F	G	H	I	J
1	First Name	Last Name	Member/Cut	OU Type	OU Name	Position	PositionDes	Years Serve	Currently in Positio	Current Term
2				Chapter	Boise Section Chapter, SSC37	Chapter Vice Chair		6.7 Yes		6.7
3				Chapter	Boise Section Chapter, ED15	Chapter Chair		5.7 Yes		5.7
4				Chapter	Boise Section Chapter, SSC37	Chapter Chair		5.1 Yes		5.1
5				Chapter	Boise Section Chapter, PE31	Chapter Vice Chair		3.6 Yes		3.6
6				Chapter	Boise Section Chapter, PE31	Chapter Treasurer		3.1 Yes		3.1
7				Section	Boise Section	Chair		2.7 Yes		2.7
8				Affinity	Boise Section Affinity Group,WIE	Affinity Group Chair		1.8 Yes		1.8
9				Affinity	Boise Section Affinity Group,YP	Affinity Group Chair		1.7 Yes		1.7
10				Chapter	Boise Section Chapter, PE31	Chapter Chair		1.6 Yes		1.6
11				Affinity	Boise Section Affinity Group,WIE	Affinity Group Vice Chair		1.6 Yes		1.6
12				Chapter	Boise Section Chapter, PE31	Chapter Secretary		1.6 Yes		1.6
13				Section	Boise Section	Vice Chair		0.7 Yes		0.7
14				Chapter	Boise Section Chapter,C16	Chapter Chair		3.2 Yes		0.7
15				Section	Boise Section	Secretary		0.7 Yes		0.7
16				Section	Boise Section	Treasurer		0.7 Yes		0.7
17				Joint Chapter	Boise Section Jt. Chapter,SP01/RA24	Chapter Chair		0.1 Yes		0.1
18				Joint Chapter	Boise Section Jt. Chapter,SP01/RA24	Chapter Vice Chair		0.1 Yes		0.1

ieeeeYearsInPosition.py

```
# ieeeeYearsInPosition.py
# Creates csv file showing time in years each volunteer has been in each position held.
# Uses as input two csv files downloaded from OU Analytics containing current volunteer and volunteer history data.

import csv
import sys
import codecs
import datetime

months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
          'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']

today = datetime.date.today()
invalidDate = datetime.date(1900,1,1) # Use Jan 1, 1900 to represent invalid date
durationServed = {} # Dictionary of records indicating how long member has held specified position
currentDuration = {} # Dictionary of records indicating how long member currently holds the specified position
memberName = {} # Dictionary mapping member#/position keys to member names
separator = '^' # Used to separate fields in dictionary keys and data

# Convert date string to date object
def convertDate(dateString):
    date = dateString.split('-')
    # Incoming date string should be day-month-year, e.g. 1-Jan-2021
    if (len(date) == 3):
        year = int(date[2])
        month = months.index(date[1])+1
        day = int(date[0])
        try:
            convertedDate = datetime.date(year, month, day)
        except ValueError:
            convertedDate = invalidDate
    else:
        convertedDate = invalidDate
    return convertedDate

# Build/update durationServed and memberName dictionaries from data in volunteer csv file
def processVolunteerFile(filename):
    # Open the csv file as a UTF-16 formatted file
    try:
        fh=codecs.open(filename,"rb","utf-16")
        # Process first row of the csv file for fieldnames
        header = [h.strip() for h in fh.readline().split('\t')]
        reader = csv.DictReader(fh, delimiter='\t', quotechar='\"', fieldnames=header)
    except OSError:
        print('Cannot open csv file', filename)
    else:
        # Process each line of the csv file
        # Each line will create or update a single record in durationServed and memberName dictionaries
        # Dictionaries are indexed by member#, OU type and name, and position and position description
        for row in reader:

            # Construct dictionary key used when recording duration served and member name
            # Key encodes the member number, OU type and name, and position held
            key = separator.join((row['Member/Customer Number'],
                                row['OU Type'],
                                row['OU Name'],
```

```

        row['Position'],
        row['Position Description']))

# Determine for this one record how long volunteer has been in position
startDate = convertDate(row['Position Start'])
if (startDate == invalidDate) or (startDate > today):
    startDate = today
endDate = convertDate(row['Position End'])
if (endDate == invalidDate) or (endDate > today):
    endDate = today
daysServed = (endDate - startDate).days
if (daysServed < 0):
    daysServed = 0

# Record duration served in the position, adding to existing value if necessary
if key in durationServed.keys():
    durationServed[key] = durationServed[key] + daysServed
    # print(f"Updated existing record key={key} name={memberName[key]} days={daysServed} total={durationServed[key]}")
else:
    durationServed[key] = daysServed
    # Record member name too
    firstName = row['First Name']
    lastName = row['Last Name']
    memberName[key] = f'{firstName}{separator}{lastName}'
    # print(f"Created new record key={key} name={memberName[key]} days={daysServed}")

# Print results from durationServed, memberName, and currentVolunteer dictionaries to stdout in csv format
def printResults():
    print("First Name\tLast Name\tMember/Customer Number\tOU Type\tOU Name\tPosition\t"
          "PositionDescription\tYears Served\tCurrently in Position\tCurrent Term")
    for key, daysServed in sorted(durationServed.items()):
        (memberNumber, ouType, ouName, position, positionDescription) = key.split(separator)
        # Calculate years served to one decimal place
        yearsServed = round(daysServed / 365, 1)
        (firstName, lastName) = memberName[key].split(separator)
        if key in currentDuration:
            inPositionNow = "Yes"
            currentYearsServed = round(currentDuration[key] / 365, 1)
        else:
            inPositionNow = "No"
            currentYearsServed = 0

    # Print the tab-delimited record
    print(f"{firstName}\t{lastName}\t{memberNumber}\t{ouType}\t{ouName}\t{position}\t"
          f"{positionDescription}\t{yearsServed}\t{inPositionNow}\t{currentYearsServed}")

# Uncomment the following lines to get a raw dump of each dictionary
# print(f'memberName={memberName}')
# print(f'durationServed={durationServed}')
# print(f'currentVolunteer={currentVolunteer}')

def showUsage():
    programName = sys.argv[0]
    print(f'python {programName} <volunteerData-csv-file> <volunteerHistoryData-csv-file>')
    print('Creates csv file showing time in years each volunteer has been in each position held.')

# Process the two volunteer csv files to output a csv file showing time each volunteer has been in each position they've held

```

```
def runIeeeYearsInPosition():
    # Command-line argument#1 is the name of the volunteer csv file
    # Command-line argument#2 is the name of the volunteer history csv file
    if len(sys.argv) != 3:
        showUsage()
    else:
        currentVolunteersFile = sys.argv[1]
        previousVolunteersFile = sys.argv[2]

        # Process current volunteers csv file
        processVolunteerFile(currentVolunteersFile)

        # Save how long current volunteers have currently been in that position
        for key in durationServed:
            currentDuration[key] = durationServed[key]

        # Process volunteer history csv file
        processVolunteerFile(previousVolunteersFile)

        # Print results
        printResults()

# Run the script.
if __name__ == '__main__':
    runIeeeYearsInPosition()
```