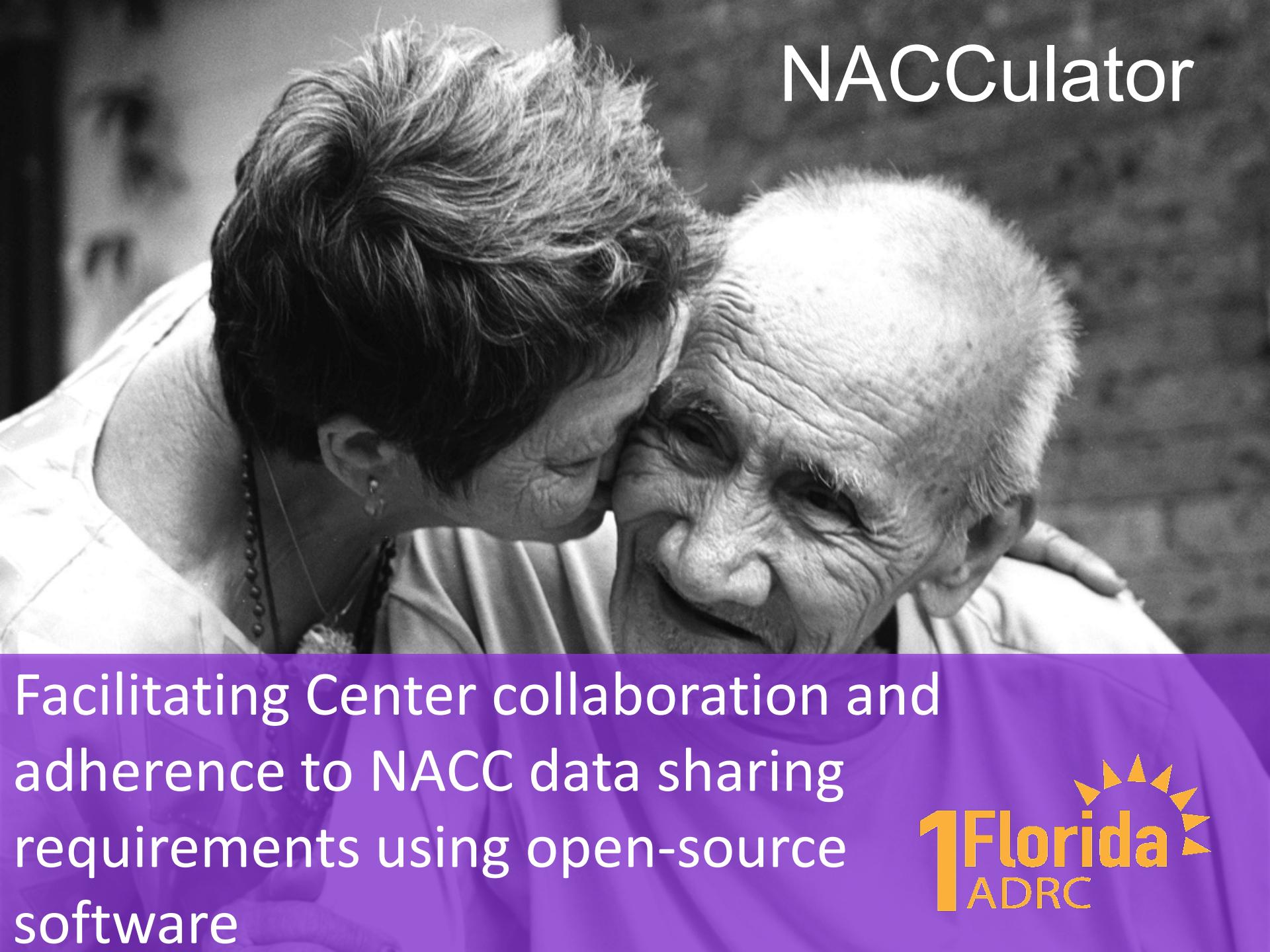




Of the 5.5 million individuals in the U.S. who live with Alzheimer's disease

**NEARLY 10% LIVE IN FLORIDA**

Costing the state  
> \$20 billion a year



# NACCulator

Facilitating Center collaboration and adherence to NACC data sharing requirements using open-source software



# Talking Points:

- The Software
- Collaboration

# THE SOFTWARE



# NACCulator

- NACCulator: a translator from CSV to fixed width format
- Written in Python
- <https://github.com/ctsit/nacculator>

```
def header_fields():  
    fields = {}  
    fields['PACKET'] = nacc.ud3.Field(name='PACKET', typename='Char', position=(1, 2), l  
    fields['FORMID'] = nacc.ud3.Field(name='FORMID', typename='Char', position=(4, 6), l  
    fields['FORMVER'] = nacc.ud3.Field(name='FORMVER', typename='Num', position=(8, 10),  
    fields['ADCID'] = nacc.ud3.Field(name='ADCID', typename='Num', position=(12, 13), le  
    fields['PTID'] = nacc.ud3.Field(name='PTID', typename='Char', position=(15, 24), len  
    fields['VISITMO'] = nacc.ud3.Field(name='VISITMO', typename='Num', position=(26, 27)  
    fields['VISITDAY'] = nacc.ud3.Field(name='VISITDAY', typename='Num', position=(29, 3  
    fields['VISITYR'] = nacc.ud3.Field(name='VISITYR', typename='Num', position=(32, 35)  
    fields['VISITNUM'] = nacc.ud3.Field(name='VISITNUM', typename='Char', position=(37,  
    fields['INITIALS'] = nacc.ud3.Field(name='INITIALS', typename='Char', position=(41,
```

<http://1floridaadrc.org/nacculator.html>



[DONATE](#) [English](#)

[Home](#) [Learn](#) [Participate](#) [Resources](#) [For Scientists](#) [Team](#) [News](#) [Contact](#)

## NACCulator

### What is NACCulator?

NACCulator is software developed to convert REDCap's comma separated value (CSV) format into fixed width format for upload to the National Alzheimer's Coordinating Center's (NACC) submission system. The 1Florida ADRC utilizes REDCap ([www.project-redcap.org](http://www.project-redcap.org)) to deliver and capture the NACC's Unified Data Set.

### Why should I use NACCulator?

NACCulator is developed in the Python programming language and is used to extract, format and upload to the NACC's submission system. The NACCulator software performs data quality checks based on the rules used in the NACC's submission system. NACCulator is reliable and repeatable.

### Where can I get NACCulator?

NACCulator is a free and open source software developed by the Clinical and Translational Science Institute - Informatics and Technology division. If you would like to utilize NACCulator in your upload and submission processes, please click the link below which will take you to an industry standard software sharing site on GitHub. There you will find instructions and resources to install and use the tool. If you have further questions about how to use the software please email [info@1floridadrc.org](mailto:info@1floridadrc.org).

[CLICK HERE TO USE THE NACCULATOR](#)

View README.md

### NACCulator

DOI: [10.5281/zenodo.46122](https://doi.org/10.5281/zenodo.46122)

Converts a CSV data file exported from REDCap into the NACC's USG3 fixed-width format.

**Files**

This is not exhaustive, but here is an explanation of some important files.

- `nacc`: top-level Python package for all things NACC.
- `nacc/redcap2nacc.py`: converts a CSV data file exported from REDCap into NACC's USG3 fixed-width format.
- `nacc/npy2xval.py`: expectation library for "Missing Values".
- `nacc/usg3/form.py`: USG3 API forms represented as Python classes.
- `usg3/generator.py`: generates Python objects based on NACC Data Element Dictionaries in CSV.

**HOWTO Convert from REDCap to NACC**

Once the project data is exported from REDCap to the CSV file `data.csv`, run:

```
$ pip install nacculator
$ nacc2nacc < data.csv > data.nacc
```

Or, if you're using the source code:

```
$ PYTHONPATH=~/nacc2nacc/nacc $ nacc2nacc < data.csv > data.nacc
```

Note: output is written to `stdout`; errors are written to `stderr`. Input can be `stdin` or the first argument passed to `nacc2nacc`.

If there are no errors, then submit the `data.nacc` file to NACC.



```

class FormA1(nacc.ud3.FieldBag):
    def __init__(self):
        self.fields = header_fields()
        self.fields['REASON'] = nacc.ud3.Field(name='REASON', typename='Num', position=(45, 45), length=1, inclusive_range=(1, 4), allowable_values=['4', '2', '1', '9'], blanks=[])
        self.fields['REFERSC'] = nacc.ud3.Field(name='REFERSC', typename='Num', position=(47, 47), length=1, inclusive_range=(1, 6), allowable_values=['8', '9', '3', '2', '1', '6'])
        self.fields['LEARNED'] = nacc.ud3.Field(name='LEARNED', typename='Num', position=(49, 49), length=1, inclusive_range=(1, 4), allowable_values=['3', '2', '1', '9', '8', '4'])
        self.fields['PRESTAT'] = nacc.ud3.Field(name='PRESTAT', typename='Num', position=(51, 51), length=1, inclusive_range=(1, 3), allowable_values=['3', '2', '1'], blanks=[])
        self.fields['PRESPART'] = nacc.ud3.Field(name='PRESPART', typename='Num', position=(53, 53), length=1, inclusive_range=(1, 2), allowable_values=['2', '1'], blanks=[])
        self.fields['SOURCENW'] = nacc.ud3.Field(name='SOURCENW', typename='Num', position=(55, 55), length=1, inclusive_range=(1, 2), allowable_values=['2', '1'], blanks=[])
        self.fields['BIRTHMO'] = nacc.ud3.Field(name='BIRTHMO', typename='Num', position=(57, 58), length=2, inclusive_range=(1, 12), allowable_values=[], blanks=[])
        self.fields['BIRTHYR'] = nacc.ud3.Field(name='BIRTHYR', typename='Num', position=(60, 63), length=4, inclusive_range=(1875, CURRENT_YEAR-15), allowable_values=[], blanks=[])
        self.fields['SEX'] = nacc.ud3.Field(name='SEX', typename='Num', position=(65, 65), length=1, inclusive_range=(1, 2), allowable_values=['2', '1'], blanks=[])
        self.fields['HISPANIC'] = nacc.ud3.Field(name='HISPANIC', typename='Num', position=(67, 67), length=1, inclusive_range=(0, 1), allowable_values=['9', '1', '0'], blanks=[])
        self.fields['HISPOR'] = nacc.ud3.Field(name='HISPOR', typename='Num', position=(69, 70), length=2, inclusive_range=(1, 6), allowable_values=['50', '99', '3', '2', '1', '6'])
        self.fields['HISPORX'] = nacc.ud3.Field(name='HISPORX', typename='Char', position=(72, 131), length=60, inclusive_range=None, allowable_values=[], blanks=['Blank if Quest...'])
        self.fields['RACE'] = nacc.ud3.Field(name='RACE', typename='Num', position=(133, 134), length=2, inclusive_range=(1, 5), allowable_values=['99', '3', '2', '1', '50', '5'])
        self.fields['RACEX'] = nacc.ud3.Field(name='RACEX', typename='Char', position=(136, 195), length=60, inclusive_range=None, allowable_values=[], blanks=['Blank if Question...'])
        self.fields['RACESEC'] = nacc.ud3.Field(name='RACESEC', typename='Num', position=(197, 198), length=2, inclusive_range=(1, 5), allowable_values=['88', '0', '99', '3', '2'])
        self.fields['RACESECX'] = nacc.ud3.Field(name='RACESECX', typename='Char', position=(200, 259), length=60, inclusive_range=None, allowable_values=[], blanks=['Blank if Quest...'])
        self.fields['RACETER'] = nacc.ud3.Field(name='RACETER', typename='Num', position=(261, 262), length=2, inclusive_range=(1, 5), allowable_values=['88', '99', '3', '2', '1'])
        self.fields['RACETERX'] = nacc.ud3.Field(name='RACETERX', typename='Char', position=(264, 323), length=60, inclusive_range=None, allowable_values=[], blanks=['Blank if Quest...'])
        self.fields['PRIMLANG'] = nacc.ud3.Field(name='PRIMLANG', typename='Num', position=(325, 325), length=1, inclusive_range=(1, 6), allowable_values=['8', '9', '3', '2', '1'])
        self.fields['PRIMLANX'] = nacc.ud3.Field(name='PRIMLANX', typename='Char', position=(327, 386), length=60, inclusive_range=None, allowable_values=[], blanks=['Blank if Quest...'])
        self.fields['EDUC'] = nacc.ud3.Field(name='EDUC', typename='Num', position=(388, 389), length=2, inclusive_range=(0, 36), allowable_values=['99'], blanks=[])
        self.fields['MARSTAT'] = nacc.ud3.Field(name='MARSTAT', typename='Num', position=(391, 391), length=1, inclusive_range=(1, 6), allowable_values=['9', '3', '2', '1', '6'])
        self.fields['LIVSITUAT'] = nacc.ud3.Field(name='LIVSITUAT', typename='Num', position=(393, 393), length=1, inclusive_range=(1, 6), allowable_values=['9', '3', '2', '1', '6'])
        self.fields['INDEPEND'] = nacc.ud3.Field(name='INDEPEND', typename='Num', position=(395, 395), length=1, inclusive_range=(1, 4), allowable_values=['3', '2', '1', '9', '4'])
        self.fields['RESIDENC'] = nacc.ud3.Field(name='RESIDENC', typename='Num', position=(397, 397), length=1, inclusive_range=(1, 4), allowable_values=['3', '2', '1', '9', '4'])
        self.fields['ZIP'] = nacc.ud3.Field(name='ZIP', typename='Char', position=(399, 401), length=3, inclusive_range=(6, 999), allowable_values=[], blanks=['Blank if Question...'])
        self.fields['HANDED'] = nacc.ud3.Field(name='HANDED', typename='Num', position=(403, 403), length=1, inclusive_range=(1, 3), allowable_values=['3', '2', '1', '9'], blanks=[])
        
class FormA2(nacc.ud3.FieldBag):
    def __init__(self):
        self.fields = header_fields()
        self.fields['INBIRMO'] = nacc.ud3.Field(name='INBIRMO', typename='Num', position=(45, 46), length=2, inclusive_range=(1, 12), allowable_values=['99'], blanks=[])
        self.fields['INBIRYR'] = nacc.ud3.Field(name='INBIRYR', typename='Num', position=(48, 51), length=4, inclusive_range=(1875, CURRENT_YEAR-15), allowable_values=['9999'], blanks=[])
        self.fields['INSEX'] = nacc.ud3.Field(name='INSEX', typename='Num', position=(53, 53), length=1, inclusive_range=(1, 2), allowable_values=['2', '1'], blanks=[])
        self.fields['INHISP'] = nacc.ud3.Field(name='INHISP', typename='Num', position=(55, 55), length=1, inclusive_range=(0, 1), allowable_values=['9', '1', '0'], blanks=[])
        self.fields['INHISPOR'] = nacc.ud3.Field(name='INHISPOR', typename='Num', position=(57, 58), length=2, inclusive_range=(1, 6), allowable_values=['50', '99', '3', '2', '1'])
        self.fields['INHISPOX'] = nacc.ud3.Field(name='INHISPOX', typename='Char', position=(60, 119), length=60, inclusive_range=None, allowable_values=[], blanks=['Blank if Quest...'])
        self.fields['INRACE'] = nacc.ud3.Field(name='INRACE', typename='Num', position=(121, 122), length=2, inclusive_range=(1, 5), allowable_values=['99', '3', '2', '1', '50'])
        self.fields['INRACEX'] = nacc.ud3.Field(name='INRACEX', typename='Char', position=(124, 183), length=60, inclusive_range=None, allowable_values=[], blanks=['Blank if Quest...'])
        self.fields['INRASEC'] = nacc.ud3.Field(name='INRASEC', typename='Num', position=(185, 186), length=2, inclusive_range=(1, 5), allowable_values=['88', '99', '3', '2', '1'])
        self.fields['INRASECX'] = nacc.ud3.Field(name='INRASECX', typename='Char', position=(188, 247), length=60, inclusive_range=None, allowable_values=[], blanks=['Blank if Quest...'])
        self.fields['INRATER'] = nacc.ud3.Field(name='INRATER', typename='Num', position=(249, 250), length=2, inclusive_range=(1, 5), allowable_values=['88', '99', '3', '2', '1'])
        self.fields['INRATERX'] = nacc.ud3.Field(name='INRATERX', typename='Char', position=(252, 311), length=60, inclusive_range=None, allowable_values=[], blanks=['Blank if Quest...'])
        self.fields['INEDUC'] = nacc.ud3.Field(name='INEDUC', typename='Num', position=(313, 314), length=2, inclusive_range=(0, 36), allowable_values=['99'], blanks=[])
        self.fields['INRELTO'] = nacc.ud3.Field(name='INRELTO', typename='Num', position=(316, 316), length=1, inclusive_range=(1, 6), allowable_values=['3', '2', '1', '6', '5'])
        self.fields['INKNOWN'] = nacc.ud3.Field(name='INKNOWN', typename='Num', position=(318, 320), length=3, inclusive_range=(0, 120), allowable_values=['999'], blanks=[])
        self.fields['INLIVWTH'] = nacc.ud3.Field(name='INLIVWTH', typename='Num', position=(322, 322), length=1, inclusive_range=(0, 1), allowable_values=['1', '0'], blanks=[])
        self.fields['INVISITS'] = nacc.ud3.Field(name='INVISITS', typename='Num', position=(324, 324), length=1, inclusive_range=(1, 6), allowable_values=['3', '2', '1', '6', '5'])
        self.fields['INCALLS'] = nacc.ud3.Field(name='INCALLS', typename='Num', position=(326, 326), length=1, inclusive_range=(1, 6), allowable_values=['3', '2', '1', '6', '5'])

```



# NACCulator

- Switches
  - -h Help
  - -fvp Follow-up Visit Packet
  - -ivp Initial Visit Packet
  - -np Neuropathology Packet
  - -f Filters
- Filters
  - Clean PTID
  - Replace Drug IDs
  - Fix Headers
  - Fill default values
  - Remove PTID
  - Get PTID

# NACCulator

- Updates
  - Python 3
  - Milestones
  - Telephone Visit Packets
  - FTLD Packets
  - LBD Packets
  - Z1X form exclusion detection

# How it Works

- Install NACCulator:
  - pip install nacculator
  - git clone git@github.com:ctsit/nacculator.git
- curl -v -d token=123456 -d content=record -d format=csv -d type=flat https://redcap.ctsi.ufl.edu/redcap/api/ > data.csv
- Run nacculator: redcap2nacc –ivp < data.csv > data.txt
- Upload to NACC website

**1Florida ADRC**

**Upload Data Files to the Working Database**

---

File to upload:  No file chosen

**Upload!**

---

**Allowable file extensions:**

.txt - Space separated text (ASCII)  
.csv - Comma separated variables  
.tsv - Tab separated variables  
.sas7bdat - SAS data file



# NACCulator Demonstration



# COLLABORATION



# Working Together

- Michigan ADRC
  - Hiroko Dodge
  - Sherry Teboe
  - Ari Bhaumik
  - Nicolas May
- Nicolas May
  - Met February of 2018
  - Conduct quarterly calls for feature updates
  - Review pull requests from GitHub

# Open-Source Software

- DxSter for algorithmic diagnosis of Alzheimer's disease

<https://ctsit.github.io/dxster/>

- NACCulator for REDCap to NACC data transfer

<https://github.com/ctsit/nacculator>

- Dropper for multi-site large file transfer

<https://github.com/ctsit/redi-dropper-client>

The image displays three screenshots of open-source software:

- DxSter:** A web application for algorithmic diagnosis of Alzheimer's disease. It features a purple calculator icon and a brief description of the DxSter algorithm.
- NACCulator:** A tool for converting CSV data from REDCap into NACC's UDS3 fixed-width format. It includes a DOI link (10.5281/zenodo.46333), a section for important files, and a screenshot of a browser showing a red header bar labeled "Redi Dropper".
- RED-I Dropper:** A web-based application for secure data transport. It shows a photograph of several glass dropper bottles and a brief description of its purpose.



# GitHub

- <https://github.com/ctsit/nacculator>
- GitHub Account
- Watch – When you watch a repository, you get notifications for new pull requests and issues.
- Star – Allows you to keep track of projects that you find interesting.
- Fork – When you fork a repository you create a copy of the repo for yourself to work on.
- Issues – Submit issues to address errors found while using the software.
- Pull requests – Submit code changes for review.

# Zenodo

- <http://doi.org/10.5281/zenodo.2875790>
- Tarun Gupta Akirala, Taeber Rapczak, Christopher P. Barnes, Matthew McConnell, Ajantha Ramineni, Kevin Hanson, Naomi Braun & L. D. Nicolas May. (2019, May 17). ctsit/nacculator: 0.6.1 (Version 0.6.1). Zenodo. <http://doi.org/10.5281/zenodo.2875790>

# NACCulator

DOI [10.5281/zenodo.2875790](https://doi.org/10.5281/zenodo.2875790)

Converts a CSV data file exported from REDCap into the NACC's UDS3 fixed-width format.

## Note

NACCulator uses Python 2.

If you are having trouble with Cappy, you may need to clone [the repo](#) and then install it from your local instance using `pip install -e <local/path/to/cappy>`

## Files

This is not exhaustive, but here is an explanation of some important files.

- `nacc/` : top-level Python package for all things NACC.
- `nacc/redcap2nacc.py` : converts a CSV data file exported from REDCap into NACC's UDS3 fixed-width format.
- `nacc/uds3/blanks.py` : specialized library for "Blanking Rules".
- `nacc/uds3/ivp/forms.py` : UDS3 IVP forms represented as Python classes.
- `tools/generator.py` : generates Python objects based on NACC Data Element Dictionaries in CSV.
- `/run_filters.py` and `run_filters.sh` : pulls data from REDCap based on the settings found in `nacculator_cfg.ini` (for `.py`) and `filters_config.cfg` (for `.sh`).



KEVIN HANSON

[kshanson@ufl.edu](mailto:kshanson@ufl.edu)

<http://1floridaadrc.org>



NICOLAS MAY

[ldmay@med.umich.edu](mailto:ldmay@med.umich.edu)

<https://alzheimers.med.umich.edu>



**Thank You**