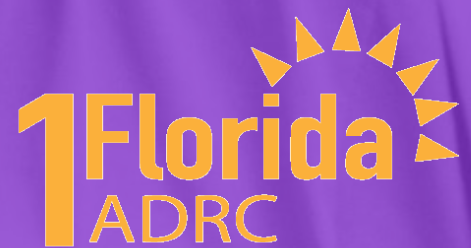


[illegible]

Costing the state
> \$20 billion a year



NACCulator: A translator for CSV files to NACC



Talking Points:

- Introduction
- Business Case
- The Software
- Collaboration



KEVIN HANSON

Co-Investigator

kshanson@ufl.edu

<http://1floridaadrc.org>

BUSINESS CASE



Business Case

- What is the problem we are trying to solve?
 - NACC input format is fixed length (position matters)
 - Need a platform to reliably apply the rule set (450+)
 - Upload data to NACC's data system
- How can we address the problem?
 - Use NACC's web entry system
 - Use SAS
 - **Write software**
- What value is generated from this option?
 - Transfer data consistently to NACC
 - Pre-check data issues prior to upload
 - Scheduled data export and upload
 - Repeatable programmable way that can be modified as changes happen

Challenges

- Tightly controlled variable definitions
- Data validation rules that may change
- Form changes

THE SOFTWARE



Authors and Designers

- Tarun Akirala
- Christopher Barnes
- Naomi Braun
- Kevin Hanson
- Matthew McConnell
- Ajantha Ramineni
- Taeber Rapczak

NACCulator

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

```
def header_fields():  
    fields = {}  
    fields['PACKET'] = nacc.uds3.Field(name='PACKET', typename='Char', position=(1, 2), l  
    fields['FORMID'] = nacc.uds3.Field(name='FORMID', typename='Char', position=(4, 6), l  
    fields['FORMVER'] = nacc.uds3.Field(name='FORMVER', typename='Num', position=(8, 10),  
    fields['ADCID'] = nacc.uds3.Field(name='ADCID', typename='Num', position=(12, 13), le  
    fields['PTID'] = nacc.uds3.Field(name='PTID', typename='Char', position=(15, 24), len  
    fields['VISITMO'] = nacc.uds3.Field(name='VISITMO', typename='Num', position=(26, 27)  
    fields['VISITDAY'] = nacc.uds3.Field(name='VISITDAY', typename='Num', position=(29, 3  
    fields['VISITYR'] = nacc.uds3.Field(name='VISITYR', typename='Num', position=(32, 35)  
    fields['VISITNUM'] = nacc.uds3.Field(name='VISITNUM', typename='Char', position=(37,  
    fields['INITIALS'] = nacc.uds3.Field(name='INITIALS', typename='Char', position=(41,
```

How it Works

- Install NACCulator: `pip install nacculator`
- `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

Allowable file extensions:

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

```
$ pip install nacculator  
$ redcap2nacc < data.csv > data.txt
```

Or, if you're using the source code:

```
$ PYTHONPATH=. ./nacc/redcap2nacc.py < data.csv > data.txt
```

The program accepts two arguments `-file` and `-(ivp|fvp|np)`. Both the arguments are optional. See the python help as:

```
$ PYTHONPATH=. ./nacc/redcap2nacc.py -h  
usage: redcap2nacc.py [-h]  
                    [-fvp | -ivp | -np | -f {cleanPtid,updateField,replaceDrugId,fillDefault,fixC1S}]  
                    [-file FILE] [-meta FILTER_META]
```

Process redcap form output to nacculator.

optional arguments:

<code>-h, --help</code>	show this help message and exit
<code>-fvp</code>	Set this flag to process as fvp data
<code>-ivp</code>	Set this flag to process as ivp data
<code>-np</code>	Set this flag to process as np data
<code>-f or --filter</code>	Accepts one of {cleanPtid,updateField,replaceDrugId,fillDefault,fixC1S}
	Set this flag to process the filter
<code>-file FILE</code>	Path of the csv file to be processed.
<code>-meta FILTER_META</code>	Input file for the filter metadata (in case cleanPtid is used)

class FormA1(nacc.uds3.FieldBag):

```
def __init__(self):
    self.fields = header_fields()
    self.fields['REASON'] = nacc.uds3.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.uds3.Field(name='REFERSC', typename='Num', position=(47, 47), length=1, inclusive_range=(1, 6), allowable_values=['8', '9', '3', '2', '1', '6'], blanks=[])
    self.fields['LEARNED'] = nacc.uds3.Field(name='LEARNED', typename='Num', position=(49, 49), length=1, inclusive_range=(1, 4), allowable_values=['3', '2', '1', '9', '8', '4'], blanks=[])
    self.fields['PRESTAT'] = nacc.uds3.Field(name='PRESTAT', typename='Num', position=(51, 51), length=1, inclusive_range=(1, 3), allowable_values=['3', '2', '1'], blanks=[])
    self.fields['PRESPART'] = nacc.uds3.Field(name='PRESPART', typename='Num', position=(53, 53), length=1, inclusive_range=(1, 2), allowable_values=['2', '1'], blanks=[])
    self.fields['SOURCENW'] = nacc.uds3.Field(name='SOURCENW', typename='Num', position=(55, 55), length=1, inclusive_range=(1, 2), allowable_values=['2', '1'], blanks=[])
    self.fields['BIRTHMO'] = nacc.uds3.Field(name='BIRTHMO', typename='Num', position=(57, 58), length=2, inclusive_range=(1, 12), allowable_values=[], blanks=[])
    self.fields['BIRTHYR'] = nacc.uds3.Field(name='BIRTHYR', typename='Num', position=(60, 63), length=4, inclusive_range=(1875, CURRENT_YEAR-15), allowable_values=[], blanks=[])
    self.fields['SEX'] = nacc.uds3.Field(name='SEX', typename='Num', position=(65, 65), length=1, inclusive_range=(1, 2), allowable_values=['2', '1'], blanks=[])
    self.fields['HISPANIC'] = nacc.uds3.Field(name='HISPANIC', typename='Num', position=(67, 67), length=1, inclusive_range=(0, 1), allowable_values=['9', '1', '0'], blanks=[])
    self.fields['HISPOR'] = nacc.uds3.Field(name='HISPOR', typename='Num', position=(69, 70), length=2, inclusive_range=(1, 6), allowable_values=['50', '99', '3', '2', '1', '6'], blanks=[])
    self.fields['HISPORX'] = nacc.uds3.Field(name='HISPORX', typename='Char', position=(72, 131), length=60, inclusive_range=None, allowable_values=[], blanks=['Blank if Questioned'])
    self.fields['RACE'] = nacc.uds3.Field(name='RACE', typename='Num', position=(133, 134), length=2, inclusive_range=(1, 5), allowable_values=['99', '3', '2', '1', '50', '5', '1', '6'], blanks=[])
    self.fields['RACEX'] = nacc.uds3.Field(name='RACEX', typename='Char', position=(136, 195), length=60, inclusive_range=None, allowable_values=[], blanks=['Blank if Questioned'])
    self.fields['RACESEC'] = nacc.uds3.Field(name='RACESEC', typename='Num', position=(197, 198), length=2, inclusive_range=(1, 5), allowable_values=['88', '0', '99', '3', '2', '1', '6'], blanks=[])
    self.fields['RACESECX'] = nacc.uds3.Field(name='RACESECX', typename='Char', position=(200, 259), length=60, inclusive_range=None, allowable_values=[], blanks=['Blank if Questioned'])
    self.fields['RACETER'] = nacc.uds3.Field(name='RACETER', typename='Num', position=(261, 262), length=2, inclusive_range=(1, 5), allowable_values=['88', '99', '3', '2', '1', '6'], blanks=[])
    self.fields['RACETERX'] = nacc.uds3.Field(name='RACETERX', typename='Char', position=(264, 323), length=60, inclusive_range=None, allowable_values=[], blanks=['Blank if Questioned'])
    self.fields['PRIMLANG'] = nacc.uds3.Field(name='PRIMLANG', typename='Num', position=(325, 325), length=1, inclusive_range=(1, 6), allowable_values=['8', '9', '3', '2', '1', '6'], blanks=[])
    self.fields['PRIMLANGX'] = nacc.uds3.Field(name='PRIMLANGX', typename='Char', position=(327, 386), length=60, inclusive_range=None, allowable_values=[], blanks=['Blank if Questioned'])
    self.fields['EDUC'] = nacc.uds3.Field(name='EDUC', typename='Num', position=(388, 389), length=2, inclusive_range=(0, 36), allowable_values=['99'], blanks=[])
    self.fields['MARISTAT'] = nacc.uds3.Field(name='MARISTAT', typename='Num', position=(391, 391), length=1, inclusive_range=(1, 6), allowable_values=['9', '3', '2', '1', '6'], blanks=[])
    self.fields['LIVSITUA'] = nacc.uds3.Field(name='LIVSITUA', typename='Num', position=(393, 393), length=1, inclusive_range=(1, 6), allowable_values=['9', '3', '2', '1', '6'], blanks=[])
    self.fields['INDEPEND'] = nacc.uds3.Field(name='INDEPEND', typename='Num', position=(395, 395), length=1, inclusive_range=(1, 4), allowable_values=['3', '2', '1', '9', '4', '5', '6'], blanks=[])
    self.fields['RESIDENC'] = nacc.uds3.Field(name='RESIDENC', typename='Num', position=(397, 397), length=1, inclusive_range=(1, 4), allowable_values=['3', '2', '1', '9', '4', '5', '6'], blanks=[])
    self.fields['ZIP'] = nacc.uds3.Field(name='ZIP', typename='Char', position=(399, 401), length=3, inclusive_range=(6, 999), allowable_values=[], blanks=['Blank if Questioned'])
    self.fields['HANDED'] = nacc.uds3.Field(name='HANDED', typename='Num', position=(403, 403), length=1, inclusive_range=(1, 3), allowable_values=['3', '2', '1', '9'], blanks=[])
```

class FormA2(nacc.uds3.FieldBag):

```
def __init__(self):
    self.fields = header_fields()
    self.fields['INBIRMO'] = nacc.uds3.Field(name='INBIRMO', typename='Num', position=(45, 46), length=2, inclusive_range=(1, 12), allowable_values=['99'], blanks=[])
    self.fields['INBIRYR'] = nacc.uds3.Field(name='INBIRYR', typename='Num', position=(48, 51), length=4, inclusive_range=(1875, CURRENT_YEAR-15), allowable_values=['9999'], blanks=[])
    self.fields['INSEX'] = nacc.uds3.Field(name='INSEX', typename='Num', position=(53, 53), length=1, inclusive_range=(1, 2), allowable_values=['2', '1'], blanks=[])
    self.fields['INHISP'] = nacc.uds3.Field(name='INHISP', typename='Num', position=(55, 55), length=1, inclusive_range=(0, 1), allowable_values=['9', '1', '0'], blanks=[])
    self.fields['INHISPOR'] = nacc.uds3.Field(name='INHISPOR', typename='Num', position=(57, 58), length=2, inclusive_range=(1, 6), allowable_values=['50', '99', '3', '2', '1', '6'], blanks=[])
    self.fields['INHISPOX'] = nacc.uds3.Field(name='INHISPOX', typename='Char', position=(60, 119), length=60, inclusive_range=None, allowable_values=[], blanks=['Blank if Questioned'])
    self.fields['INRACE'] = nacc.uds3.Field(name='INRACE', typename='Num', position=(121, 122), length=2, inclusive_range=(1, 5), allowable_values=['99', '3', '2', '1', '50', '5', '1', '6'], blanks=[])
    self.fields['INRACEX'] = nacc.uds3.Field(name='INRACEX', typename='Char', position=(124, 183), length=60, inclusive_range=None, allowable_values=[], blanks=['Blank if Questioned'])
    self.fields['INRASEC'] = nacc.uds3.Field(name='INRASEC', typename='Num', position=(185, 186), length=2, inclusive_range=(1, 5), allowable_values=['88', '99', '3', '2', '1', '6'], blanks=[])
    self.fields['INRASECX'] = nacc.uds3.Field(name='INRASECX', typename='Char', position=(188, 247), length=60, inclusive_range=None, allowable_values=[], blanks=['Blank if Questioned'])
    self.fields['INRATER'] = nacc.uds3.Field(name='INRATER', typename='Num', position=(249, 250), length=2, inclusive_range=(1, 5), allowable_values=['88', '99', '3', '2', '1', '6'], blanks=[])
    self.fields['INRATERX'] = nacc.uds3.Field(name='INRATERX', typename='Char', position=(252, 311), length=60, inclusive_range=None, allowable_values=[], blanks=['Blank if Questioned'])
    self.fields['INEDUC'] = nacc.uds3.Field(name='INEDUC', typename='Num', position=(313, 314), length=2, inclusive_range=(0, 36), allowable_values=['99'], blanks=[])
    self.fields['INRELTO'] = nacc.uds3.Field(name='INRELTO', typename='Num', position=(316, 316), length=1, inclusive_range=(1, 6), allowable_values=['3', '2', '1', '6', '5', '1', '6'], blanks=[])
    self.fields['INKNOWN'] = nacc.uds3.Field(name='INKNOWN', typename='Num', position=(318, 320), length=3, inclusive_range=(0, 120), allowable_values=['999'], blanks=[])
    self.fields['INLIWTH'] = nacc.uds3.Field(name='INLIWTH', typename='Num', position=(322, 322), length=1, inclusive_range=(0, 1), allowable_values=['1', '0'], blanks=[])
    self.fields['INVISITS'] = nacc.uds3.Field(name='INVISITS', typename='Num', position=(324, 324), length=1, inclusive_range=(1, 6), allowable_values=['3', '2', '1', '6', '5', '1', '6'], blanks=[])
    self.fields['INCALLS'] = nacc.uds3.Field(name='INCALLS', typename='Num', position=(326, 326), length=1, inclusive_range=(1, 6), allowable_values=['3', '2', '1', '6', '5', '1', '6'], blanks=[])
```

COLLABORATION



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 overlapping screenshots of open-source software interfaces. The top-left screenshot shows the DxSter website, which features a purple header with navigation links (About, Go to DxSter, Publications, Get the Code, FAQ, Examples, Help) and a central section titled 'What is DxSter?' with a calculator icon. The top-right screenshot shows the NACCulator website, which has a white header and a section titled 'NACCulator' with a DOI link (DOI: 10.5281/zenodo.46333) and a list of files. The bottom screenshot shows the Redi Dropper website, which has a red header with a 'Login' button and a section titled 'What is RED-I Dropper?' with a photo of laboratory bottles. The Redi Dropper website also includes a footer with the text 'CTS-IT at the University of Florida · All Rights Reserved · Version 0.0.3b'.

DxSter

What is DxSter?

DxSter, the Alzheimer's disease Algorithmic Diagnostic Helper, is a codified algorithm that can diagnose normal cognition, MCI and Dementia which serves as a valid alternative that reduces time, effort and biases associated with consensus diagnosis. Codifying the AlgDx algorithmic diagnosis project is an effort to translate the algorithm that calculates the physician and clinical diagnosis. The algorithm is designed to reduce the number of cases

NACCulator

DOI: 10.5281/zenodo.46333

Converts a CSV data file exported from REDCap into the NACC's UDS3 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 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.

Publications

Duara, R., Loewenstein, D., et al. (2018). Normal Cognition, MCI and Dementia. American Association of Neuropsychiatrists.

Redi Dropper

Login

What is RED-I Dropper?

RED-I Dropper is a web-based application used to assist in the secure transport of data from one location to another. This transfer can include both sensitive and non-sensitive data.

CTS-IT at the University of Florida · All Rights Reserved · Version 0.0.3b

Acknowledgements

- Kansas ADC

- Jonathan Mahnken



- Suzanne Hunt

- Michigan ADC

- Hiroko Dodge



- Sherry Teboe
 - Sean Ma
 - Ari Bhaumik



Thank You