

---

# **smartva-dhis2 Documentation**

**David Huser**

**Sep 21, 2018**



---

## Contents

---

<b>1</b>	<b>DHIS2 configuration</b>	<b>3</b>
<b>2</b>	<b>Installation</b>	<b>5</b>
2.1	Run . . . . .	5
2.2	Tests . . . . .	6
2.3	Deployment . . . . .	6
<b>3</b>	<b>App configuration</b>	<b>9</b>
3.1	config.ini . . . . .	9
3.2	Org Unit set up . . . . .	10
3.3	Country list . . . . .	10
<b>4</b>	<b>Local SQLite database</b>	<b>17</b>
4.1	Querying and exporting . . . . .	17
4.2	Backup . . . . .	18
4.3	Schema migrations . . . . .	18
<b>5</b>	<b>DHIS2 to DHIS2 Event data transfer</b>	<b>19</b>
5.1	DHIS2 configuration . . . . .	19
5.2	Installation . . . . .	19
5.3	Script configuration . . . . .	20
5.4	Permanently delete soft deleted events . . . . .	20
5.5	Run . . . . .	21
<b>6</b>	<b>smartva-dhis2 components</b>	<b>23</b>
<b>7</b>	<b>Development</b>	<b>27</b>
7.1	Testing . . . . .	27
7.2	Profiling . . . . .	27
7.3	Debugging . . . . .	27
7.4	Releasing . . . . .	28
7.5	Commandline interface . . . . .	28
7.6	Updating Documentation . . . . .	29
<b>8</b>	<b>Changelog</b>	<b>31</b>
<b>9</b>	<b>Links</b>	<b>33</b>



A Python package for the integration of Verbal Autopsy data into DHIS2.

It downloads [ODK Aggregate](#) records via [ODK Briefcase](#), runs the [SmartVA / Tariff 2.0](#) algorithm to determine the most probable Cause of Death, transforms it to a DHIS2-compatible Program Event and posts it to DHIS2. Any data validation errors or DHIS2 import errors are written to a local SQLite database which can be queried via the command line and exported to various formats.

It also checks DHIS2 for duplicate events (by Study ID Number) before posting and auto-assigns organisation units to the program.

- Documentation: [smartva-dhis2.readthedocs.io](http://smartva-dhis2.readthedocs.io)
- Github: [github.com/D4H-VA/smartva-dhis2](https://github.com/D4H-VA/smartva-dhis2)
- Virtual Machine Demo: [Ubuntu 16.04 LTS image](#)

To transfer events to a target DHIS2 (e.g. a Ministry of Health DHIS2), see [DHIS2 to DHIS2 Event data transfer](#).



---

## DHIS2 configuration

---

All metadata in DHIS2 is required to be configured before running this application.

This process will set up the VA module in the DHIS2 server. This module is required to be set up before VA data can be pushed to DHIS2.

---

**Note:** The Organisation Unit hierarchy in DHIS2 should be aligned with the VA Questionnaire. This application assumes it is set up correctly (however it auto-assigns the Organisation Unit to the program if not already assigned).

---

In order to install metadata, import the the following metadata files with DHIS2's Import/Export app. Always do a "Dry run" first.

1. Import metadata/optionset\_age\_category.csv
2. Import metadata/optionset\_cause\_of\_death.csv
3. Import metadata/optionset\_ICD10.csv
4. Import metadata/optionset\_sex.csv
5. Import metadata/dataelements.csv
6. Import metadata/program.json
7. (Import metadata/dashboard.json)

Probably it is a good idea to create a dedicated `verbal-autopsy-bot` user account with at least the following access:

- the whole Organisation Unit hierarchy for both data capture and data analysis
- the User Role authority to send Events
- the User Role authority to access the program (read and write)
- the User Role *read* authority for all imported metadata

This has the advantage that the dedicated username shows up in DHIS 2 log files.





# CHAPTER 2

---

## Installation

---

---

**Note:** It is *highly recommended* to test the whole configuration part including running over a period of time on a development/test server before implementing it in production.

---

Use `pipenv` (the recommended wrapper for `virtualenvs` and `pip`) to install this package. It depends on Python 3.5+ and various packages as described in `Pipfile` and DHIS2 2.28 as for now.

Libraries included:

- Briefcase version: 1.10.1 Production (see [ODK Github](#))
- smartva: SmartVA-Analyze, version 2.0.0-a8

System requirements:

- Min. 2GB RAM
- Min. 2 CPUs

Ubuntu installation (tested with 16.04 LTS)

```
sudo apt update
sudo apt install python3
sudo apt install python3-venv python3-pip

(As a non-root user)
pip3 install pipenv --user
git clone --depth=1 https://github.com/D4H-VA/smartva-dhis2
cd smartva-dhis2
pipenv install --ignore-pipfile --deploy
```

## 2.1 Run

Refer to the configuration pages first before running it:

1. *DHIS2 configuration*
2. *App configuration*

```
cd ~/smartva-dhis2    (adjust to path where you cloned the repository)
pipenv run smartva-dhis2 [--options]
```

Options are:

```
--manual          Skip download of ODK aggregate file, provide local file path_
↪instead
--all             Pull ALL briefcases instead of relative time window
```

**If you do not provide any argument**, it will attempt to import ODK Aggregate records from last week (today minus 7 days). e.g. if today is 2018-04-08 it attempts to download records for 2018-04-01:00:00:00 to 2018-04-01:23:59:59.

This is scheduled to run every three hours (leading to messages that the record is already in DHIS2) but then it's expected.

---

**Note:** This application builds on the fact that *Study ID numbers* (SID) are **always unique** and **not modified in DHIS2 after the import**.

---

## 2.2 Tests

To run tests:

```
pipenv install --dev
pipenv run python setup.py test
```

## 2.3 Deployment

Make sure the script is running even after server reboots - how this is achieved depends on the Operating System. For systemd-based Operating Systems, you can install the following service.

```
[Unit]
Description=smartva-dhis2
After=multi-user.target

[Service]
Type=simple
Restart=always
User=ubuntu
WorkingDirectory=~/smartva-dhis2
ExecStart=~/local/bin/pipenv run smartva-dhis2

[Install]
WantedBy=multi-user.target
```

- Adjust ~/smartva-dhis2 to where you've installed the repository
- Adjust the path to pipenv - you can find out the path by calling `which pipenv`.

- Adjust the `ubuntu` user to the user that runs the script
- `~` means *expanding to the home folder of the user as specified in `User=`*.

Systemd service installation on Ubuntu:

```
sudo nano /etc/systemd/system/smartva-dhis2.service
(adjust and paste above config)
sudo systemctl enable smartva-dhis2.service
sudo systemctl start smartva-dhis2.service

(to see the status of the service:)
sudo systemctl start smartva-dhis2.service

(check log files:)
tail -f smartva_dhis2.log
```



### 3.1 config.ini

All configuration is defined in `config.ini`. Note that there are no apostrophes (" or ') in this file.

#### [auth]

**auth\_file** A file path to where authentication details for DHIS2 and ODK Aggregate are stored - see `dish.json` for structure of the file. Keep it on a secure place and refer to its file path.

#### [logging]

**logfile** Where the application should log to, e.g. `/var/log/smartvadhis2.log`

**level** Minimum Log level - e.g. `INFO` logs all info messages, warnings, errors. Must be one of: `DEBUG`, `INFO`, `WARNINGS`

#### [database]

**db\_queries\_log** Whether to log all local database queries as well. Either `true` or `false`.

**db\_name** Name of the local database file, e.g. `smartva-dhis2.db`

#### [odk]

**form\_id** Verbal Autopsy ODK Form ID, e.g. `SmartVA_Bangla_v7`

**sid\_regex** Regular Expression that matches a Verbal Autopsy Study ID number, e.g. `^VA_[0-9]{17}$`. Check regex with online tools, e.g. [regex101.com](http://regex101.com). If you want to allow any SID format (*not recommended*), you can put `sid_regex = .*`.

#### [smartva]

**ignore\_columns** Which CSV columns in the SmartVA CSV output to ignore for further processing. Must be delimited by commas `,` and without space, e.g. `geography1,geography2,geography4,geography5,cause34`

**algorithm\_version** With which version the CoD was obtained, e.g. `Tariff 2.0`.

**country** Data origin country abbreviation. See below for full list.

**hiv** Data is from an HIV region

**malaria** Data is from a Malaria region.

**hce** Use Health Care Experience (HCE) variables.

For more information about SmartVA options refer to the SmartVA Help: [PDF](#).

**[dhis]**

**program** The Unique Identifier (UID) of the Verbal Autopsy DHIS2 program.

**program\_stage** The Unique Identifier (UID) of the Verbal Autopsy DHIS2 program *stage*.

**api\_version** DHIS2 API Version (e.g. 28)

For further mapping details see also the `smartva/core/mapping.py` module.

## 3.2 Org Unit set up

In order to determine the location of the Verbal Autopsy, you need to define the following steps:

1. Find out where the orgUnit is located in your aggregate CSV
2. ignore certain columns in the `smartva.ignore_columns` section of `config.ini` (see above)
3. In `smartvadhiss2/core/mapping.py`, update the `csv_name` property in the `Orgunit` class.

In the example `config.ini`, the column holding the Organisation Unit UID is `geography3`.

## 3.3 Country list

See section `[smartva]` above.

Country list:

- Unknown [default]
- Afghanistan (AFG)
- Albania (ALB)
- Algeria (DZA)
- Andorra (AND)
- Angola (AGO)
- Antigua and Barbuda (ATG)
- Argentina (ARG)
- Armenia (ARM)
- Australia (AUS)
- Austria (AUT)
- Azerbaijan (AZE)
- Bahrain (BHR)
- Bangladesh (BGD)
- Barbados (BRB)

- Belarus (BLR)
- Belgium (BEL)
- Belize (BLZ)
- Benin (BEN)
- Bhutan (BTN)
- Bolivia (BOL)
- Bosnia and Herzegovina (BIH)
- Botswana (BWA)
- Brazil (BRA)
- Brunei (BRN)
- Bulgaria (BGR)
- Burkina Faso (BFA)
- Burundi (BDI)
- Cambodia (KHM)
- Cameroon (CMR)
- Canada (CAN)
- Cape Verde (CPV)
- Central African Republic (CAF)
- Chad (TCD)
- Chile (CHL)
- China (CHN)
- Colombia (COL)
- Comoros (COM)
- Congo (COG)
- Costa Rica (CRI)
- Cote d'Ivoire (CIV)
- Croatia (HRV)
- Cuba (CUB)
- Cyprus (CYP)
- Czech Republic (CZE)
- Democratic Republic of the Congo (COD)
- Denmark (DNK)
- Djibouti (DJI)
- Dominica (DMA)
- Dominican Republic (DOM)
- Ecuador (ECU)

- Egypt (EGY)
- El Salvador (SLV)
- Equatorial Guinea (GNQ)
- Eritrea (ERI)
- Estonia (EST)
- Ethiopia (ETH)
- Federated States of Micronesia (FSM)
- Fiji (FJI)
- Finland (FIN)
- France (FRA)
- Gabon (GAB)
- Georgia (GEO)
- Germany (DEU)
- Ghana (GHA)
- Greece (GRC)
- Grenada (GRD)
- Guatemala (GTM)
- Guinea (GIN)
- Guinea-Bissau (GNB)
- Guyana (GUY)
- Haiti (HTI)
- Honduras (HND)
- Hungary (HUN)
- Iceland (ISL)
- India (IND)
- Indonesia (IDN)
- Iran (IRN)
- Iraq (IRQ)
- Ireland (IRL)
- Israel (ISR)
- Italy (ITA)
- Jamaica (JAM)
- Japan (JPN)
- Jordan (JOR)
- Kazakhstan (KAZ)
- Kenya (KEN)



- Kiribati (KIR)
- Kuwait (KWT)
- Kyrgyzstan (KGZ)
- Laos (LAO)
- Latvia (LVA)
- Lebanon (LBN)
- Lesotho (LSO)
- Liberia (LBR)
- Libya (LBY)
- Lithuania (LTU)
- Luxembourg (LUX)
- Macedonia (MKD)
- Madagascar (MDG)
- Malawi (MWI)
- Malaysia (MYS)
- Maldives (MDV)
- Mali (MLI)
- Malta (MLT)
- Marshall Islands (MHL)
- Mauritania (MRT)
- Mauritius (MUS)
- Mexico (MEX)
- Moldova (MDA)
- Mongolia (MNG)
- Montenegro (MNE)
- Morocco (MAR)
- Mozambique (MOZ)
- Myanmar (MMR)
- Namibia (NAM)
- Nepal (NPL)
- Netherlands (NLD)
- New Zealand (NZL)
- Nicaragua (NIC)
- Niger (NER)
- Nigeria (NGA)
- North Korea (PRK)

- Norway (NOR)
- Oman (OMN)
- Pakistan (PAK)
- Palestine (PSE)
- Panama (PAN)
- Papua New Guinea (PNG)
- Paraguay (PRY)
- Peru (PER)
- Philippines (PHL)
- Poland (POL)
- Portugal (PRT)
- Qatar (QAT)
- Romania (ROU)
- Russia (RUS)
- Rwanda (RWA)
- Saint Lucia (LCA)
- Saint Vincent and the Grenadines (VCT)
- Samoa (WSM)
- Sao Tome and Principe (STP)
- Saudi Arabia (SAU)
- Senegal (SEN)
- Serbia (SRB)
- Seychelles (SYC)
- Sierra Leone (SLE)
- Singapore (SGP)
- Slovakia (SVK)
- Slovenia (SVN)
- Solomon Islands (SLB)
- Somalia (SOM)
- South Africa (ZAF)
- South Korea (KOR)
- Spain (ESP)
- Sri Lanka (LKA)
- Sudan (SDN)
- Suriname (SUR)
- Swaziland (SWZ)

- Sweden (SWE)
- Switzerland (CHE)
- Syria (SYR)
- Taiwan (TWN)
- Tajikistan (TJK)
- Tanzania (TZA)
- Thailand (THA)
- The Bahamas (BHS)
- The Gambia (GMB)
- Timor-Leste (TLS)
- Togo (TGO)
- Tonga (TON)
- Trinidad and Tobago (TTO)
- Tunisia (TUN)
- Turkey (TUR)
- Turkmenistan (TKM)
- Uganda (UGA)
- Ukraine (UKR)
- United Arab Emirates (ARE)
- United Kingdom (GBR)
- United States (USA)
- Uruguay (URY)
- Uzbekistan (UZB)
- Vanuatu (VUT)
- Venezuela (VEN)
- Vietnam (VNM)
- Yemen (YEM)
- Zambia (ZMB)
- Zimbabwe (ZWE)



---

## Local SQLite database

---

A single database file as specified in `config.ini` is used. The database includes three tables which gets only populated when something went wrong.

- `person`: All details regarding a person record
- `failure`: Categorization of import errors. It is automatically sourced from the code (see `exceptions` folder) upon database creation.
- `person_failure`: The linking table between a person and a failure category.

Check `smartvadis2/core/models.py` for the database schema.

If there is ever a need to move to a full-blown DBMS (e.g. Postgres, Redshift) it is hypothetically easy to switch since it relies on an ORM (Object-relational mapping) - namely `SQLAlchemy`.

### 4.1 Querying and exporting

A command-line tool to query and export from the local database is included, intended for report-style exports.

Use the `records` command (check [kennethreitz/records](#) for details)

```
pipenv run records 'select first_name, surname, cause_of_death from person' --
↳url=sqlite:///db/smartva-dhis2.db
```

would for example yield:

```
first_name|surname |cause_of_death
-----|-----|-----
A amin    |Skywalker|Homicide
Joan     |Ark      |Diabetes
Han      |Solo     |Congenital malformation
```

To export it as a CSV file:

```
pipenv run records 'select sid, surname, cause_of_death from person' csv --  
↳url=sqlite:///db/smartva-dhis2.db > export.csv
```

### 4.1.1 SQL snippets

Get the different error code count since a certain date:

```
SELECT count(pf.failureid), f.failuredescription  
FROM person_failure pf  
JOIN failure f  
ON f.failureid = pf.failureid  
WHERE pf.created >= '2018-04-24'  
GROUP BY pf.failureid;
```

Get the names and SIDs of attempted imports but were already in DHIS2 (duplicates):

```
SELECT p.first_name, p.first_name_2nd, p.surname, p.sid  
FROM person p  
JOIN person_failure pf  
ON pf.personid = p.personid  
WHERE pf.failureid = 704;
```

### 4.1.2 Supported export formats

csv, tsv, json, yaml, html, xls, xlsx, dbf, latex, ods

## 4.2 Backup

It is advised to automate a backup of the local database (which is just a file) to a secure remote location, preferably keeping old versions (instead of replacing it every time).

Standard free and open source command-line tools for backing up files remotely:

- `rsync`
- `s3cmd` for transferring to S3 compatible cloud storage service providers

## 4.3 Schema migrations

Database schema migrations - if necessary - are facilitated by [Alembic](#). Just run `pipenv alembic init alembic` and follow the instructions in the link.

---

## DHIS2 to DHIS2 Event data transfer

---

To migrate DHIS2 events from one instance to another DHIS2 instance, use the script located at [github.com/D4H-VA/smartva-dhis2-data-transfer](https://github.com/D4H-VA/smartva-dhis2-data-transfer).

Same as with the main application, it auto-assigns Organisation Units and avoids importing duplicates by assuring that no event exists already with the same Study ID number (see `study_id` below).

---

**Note:** This application builds on the fact that *Study ID numbers* (SID) are **always unique** and **not modified in DHIS2 after the import**.

---

### 5.1 DHIS2 configuration

Metadata must be aligned with the setup as described in *DHIS2 configuration* - you can import the same files.

### 5.2 Installation

```
sudo apt update
sudo apt install python3
sudo apt install python3-venv python3-pip

(As a non-root user)
pip3 install pipenv --user
git clone --depth=1 https://github.com/D4H-VA/smartva-dhis2-data-transfer
cd smartva-dhis2-data-transfer
pipenv install --ignore-pipfile --deploy
```

## 5.3 Script configuration

A similar `config.ini` file can be found in this repository.

### [auth]

**auth\_file** A file path to where authentication details for source and target DHIS2 is stored - see `dish.json` for structure of the file. Keep it on a secure place and refer to its file path.

### [dhis]

**program** The Unique Identifier (UID) of the Verbal Autopsy DHIS2 program.

**program\_stage** The Unique Identifier (UID) of the Verbal Autopsy DHIS2 program *stage*.

**study\_id** The Unique Identifier (UID) of the Data Element of the Study ID number. Should probably be `L370gG5pb3P` - the same as in *App configuration*

**attribute\_category\_option** The Unique Identifier (UID) of the Category Option to store the events for. If no special requirements are in place, it should be the `default` UID - get the UID via `<target-dhis2.org>/api/categoryOptions?filter=name:eq:default`.

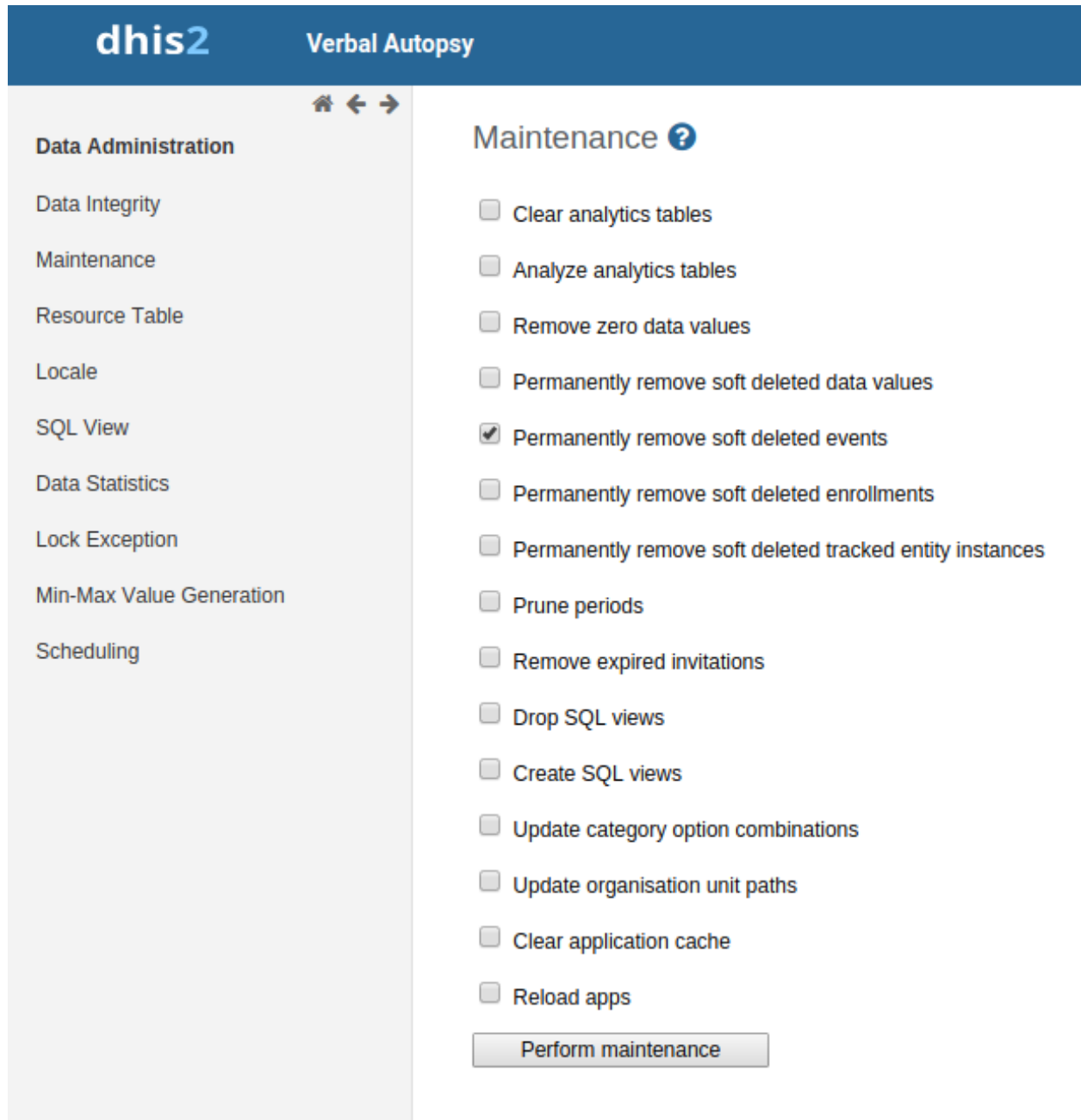
**attribute\_option\_combo** The Unique Identifier (UID) of the Category Option Combination that holds above Category Option - get the UID via `<target-dhis2.org>/api/categoryOptionCombos?filter=name:eq:default`.

**retain\_event\_uid** Remove the Event UID before posting to the target DHIS2. if `true`: keep Event UID in order to skip (the required) permanent deletion of events (see below) if `false`: delete Event UID and let the target DHIS2 assign a new Event UID for the import

## 5.4 Permanently delete soft deleted events

If `retain_event_uid` is configured to be `true` (see above), you may run into HTTP 500 Server Errors when trying to re-import events (to a DHIS2 instance running **2.26** or later) that already were imported earlier but then deleted. To permanently remove those events, you can go to Apps > Data Administration > Maintenance and tick the following box:





The screenshot shows the dhis2 Verbal Autopsy interface. The left sidebar contains a menu with the following items: Data Administration, Data Integrity, Maintenance, Resource Table, Locale, SQL View, Data Statistics, Lock Exception, Min-Max Value Generation, and Scheduling. The main content area is titled "Maintenance" and contains a list of 14 checkboxes, each followed by a description of a maintenance task. The checkbox for "Permanently remove soft deleted events" is checked. At the bottom of the list is a button labeled "Perform maintenance".

**dhis2** Verbal Autopsy

Home ← →

**Data Administration**

- Data Integrity
- Maintenance
- Resource Table
- Locale
- SQL View
- Data Statistics
- Lock Exception
- Min-Max Value Generation
- Scheduling

**Maintenance** ?

- Clear analytics tables
- Analyze analytics tables
- Remove zero data values
- Permanently remove soft deleted data values
- Permanently remove soft deleted events
- Permanently remove soft deleted enrollments
- Permanently remove soft deleted tracked entity instances
- Prune periods
- Remove expired invitations
- Drop SQL views
- Create SQL views
- Update category option combinations
- Update organisation unit paths
- Clear application cache
- Reload apps

Perform maintenance

Click “Perform maintenance”. This removes deleted events from the database (while before they were just marked as “deleted” but not really deleted).

## 5.5 Run

```
cd ~/smartva-dhis2-data-transfer (adjust to path where you cloned the repository)
pipenv run data-transfer --log=/path/to/logfile.log [--options]
```

Options are:

```
--all           Import all events of a program
--from_date     Import events of a certain date
```

If you do not provide any optional argument, it will attempt to import yesterday's events.

### 5.5.1 Cron job

This can be installed in a cron job - e.g. every day on 23:15 / 11:15 PM:

```
15 23 * * * cd /home/ubuntu/smartva-dhis2-data-transfer && /home/ubuntu/.local/bin/
↳ pipenv run data-transfer --log=/var/log/verbal_autopsies_import.log
```



smartva-dhis2 components

smartva-dhis2 components

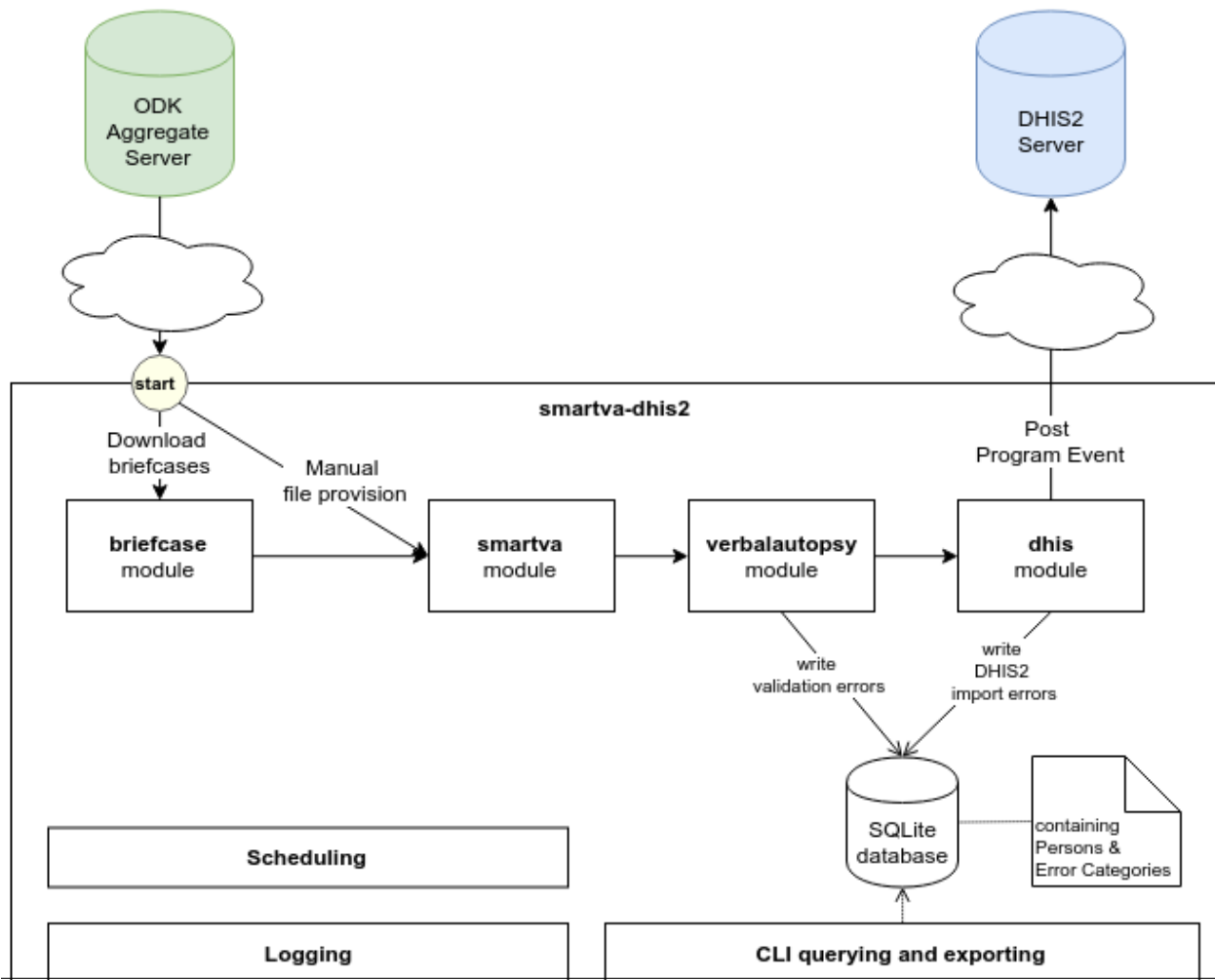


Diagram created with [draw.io](#) and `_static/components.xml`



Follow instructions in *Installation*, but install it with additional `--dev -e .` flag:

```
pipenv install --ignore-pipfile --dev -e .
```

Entry point for the code is at `__main__.py`.

### 7.1 Testing

`pytest` is used for Unit testing.

```
pipenv run python setup.py test
```

### 7.2 Profiling

To run profiles with `cProfile` (50.000 ODK records) and `/usr/bin/time` (10.000 ODK records), run:

```
pipenv run python setup.py profile
```

### 7.3 Debugging

In order to debug the module, you can create a file called `debug.py` next to `__main__.py` (with same code content) and create a debug profile in PyCharm like this:

**Name:**

**Configuration** | **Logs**

**Script path:**

**Parameters:**

**Environment**

**Environment variables:**

**Python interpreter:**

**Interpreter options:**

**Working directory:**

Add content roots to PYTHONPATH

Add source roots to PYTHONPATH

Emulate terminal in output console

Run with Python console

## 7.4 Releasing

- Use Semantic Versioning:

“Consider a version format of X.Y.Z (Major.Minor.Patch). Bug fixes not affecting the API increment the patch version, backwards compatible API additions/changes increment the minor version, and backwards incompatible API changes increment the major version.”

—<https://semver.org>

## 7.5 Commandline interface

- delete ALL events (asks first)
- export program metadata w/ dependencies
- print error categories (exceptions)



```
pipenv run smartva-dhis2-cli --help
usage:

--delete_events      Delete all events
--download_program_metadata
                    Download DHIS2 program metadata
--print_error_categories
                    Print error categories inserted into the database
```

## 7.6 Updating Documentation

- add RestructuredText files (like this one) to docs and link them in index.rst
- pipenv shell, then cd docs and finally make html



# CHAPTER 8

---

## Changelog

---

### **v0.2.1**

- moving on from demo schedules (every 30 seconds) to production schedules (every 3 hours)

### **v0.2.0**

- added scheduling
- removed pip installation
- get root orgunit automatically
- auto-assign orgunits
- updated smartva to 2.0.0a8
- updated and included Briefcase to 1.10.1
- adjust timewindow to be just one day
- prettier logs

### **v0.1.0**

- initial prototype version



## CHAPTER 9

---

### Links

---

- [CRVS Knowledge Gateway](#)
- [DHIS 2](#)
- [IHME / SmartVA](#)