
VA Explorer Documentation

Release 1.0.1

MITRE

2023-10-02

USAGE

I	Usage	2
1	Getting Started	4
1.1	Installation & Setup	4
1.1.1	Obtain a server (either your own hardware or cloud provider)	4
1.1.2	Obtain a web address/domain name	5
1.1.3	Prepare server to host VA Explorer	5
1.1.4	Deploy and run VA Explorer via Docker	5
1.1.5	Create your first Admin account and log in	6
1.2	Configuration & Deployment	6
1.3	Tips & Best Practices	8
1.3.1	Setting up a mail server	9
1.3.2	Implementing server monitoring	9
1.3.3	Configuring things for better security	9
2	Features	11
2.1	Tablet Friendly Interface	11
2.2	User Identity & Access Management	12
2.3	VA Data Import	13
2.4	Data Collection Supervision	13
2.5	Processing & Analyzing of Cause of Death Data	14
2.6	Working with VA Questionnaires	15
2.7	Processed Data Export	15
3	Integrations	17
3.1	Networking	17
3.2	ODK Central	18
3.3	KoboToolbox	19
3.4	DHIS2	19
3.5	Algorithm Support	20
3.5.1	pyCrossVA	20
3.5.2	InterVA5	21
II	Guides	22
4	General Information	24
4.1	The VA Questionnaire	24
4.2	Your VA Explorer Account	24
4.2.1	User Permissions	25
4.2.2	Common Actions Across User Roles	26

5	User Roles	30
5.1	Data Managers	30
5.1.1	Viewing Users & Supervision Information	30
5.1.2	Exporting Data	31
5.1.3	Running Data Cleanup Operations	31
5.1.4	Repairing VA Errors & Warnings	32
5.2	Data Viewers	32
5.3	Field Workers	32
5.3.1	Partially Repairing VA Errors & Warnings	33
6	The Admin Role	34
6.1	User Admin	34
6.1.1	Loading Locations for Geographic Access Support	34
6.1.2	Creating & Editing Users	35
6.1.3	Deactivating Users	36
6.1.4	Bulk Creating Users	36
6.2	Data Admin	36
6.2.1	Manually Importing Data	36
6.2.2	Manually Running Coding Algorithms	36
6.3	Management Commands	36
7	IT Support	39
7.1	Levels of Support Recommended for VA Explorer	39
7.2	Backing Up VA Explorer	40
7.3	Upgrading VA Explorer	40
8	Troubleshooting	42
8.1	Reading and Interpreting VA Explorer Logs	42
8.2	Frequently Asked Questions	42
III	Development	45
9	Development	47
9.1	Setting Up & Building	47
9.2	Testing & Running Locally	48
9.3	Running Integrations Locally	49
9.4	Development Commands	49
10	Extras	51
10.1	Security & Privacy	51
10.2	Attribution	51
10.3	Copyright & License	51
10.4	Contact	52
10.5	Other VA Resources	52
10.6	Glossary	52
Index		54



Fig. 1: An example VA Interview Image from healthdata.org/data-tools-practices/verbal-autopsy

Verbal autopsies (*VAs*) are a World Health Organization (*WHO*) -standardized tool for conducting structured interviews to determine an individual's most likely cause of death (*CoD*). *VAs* use information from next of kin, or witnesses, regarding symptoms and health history of the deceased prior to their death.

VA Explorer is an open-source web application built to help individuals and organizations support the management, analysis, and dissemination of *VAs*. VA Explorer aims to support integration with existing civil registration and vital statistics (*CRVS*) systems at scale while simplifying the use of *VAs* for those individuals and organizations.

This documentation aims to be a universal reference for everything related to VA Explorer. For example:

- If you are interested in using VA Explorer immediately, *Getting Started* (page 4) walks users through installing and running the service.
- For evaluating whether VA Explorer meets your or your organization's needs, more information on supported features can be found in *Features* (page 11) while more info on the *CRVS* systems VA Explorer works with can be found in *Integrations* (page 17)
- Those running an instance of VA Explorer already who would like to become more familiar with what they can do in their system role as an Admin, Data Manager, Data Viewer, or Field Worker can do so in *User Training* (page 30) or *Admin Training* (page 34)
- Users wishing to report bugs, request features, or developers wishing to contribute to the VA Explorer effort can find resources in *Development* (page 47)

Contents:

Part I

Usage

GETTING STARTED

New to VA Explorer and ready to get it up and running? Start here. See Also: If you already have a VA Explorer instance, check our *IT Support Guides* (page 39)

1.1 Installation & Setup

VA Explorer is distributed and installed using Docker. Docker supports near automatic configuration and installation of VA Explorer no matter where you choose to install it. In general, you can expect an installation and hosting of VA Explorer to involve these steps:

1. Obtain a server (either your own hardware or cloud provider)
2. Obtain a web address/domain name
3. Prepare server to host VA Explorer
4. Deploy and run VA Explorer via Docker
5. Create your first Admin account and log in

Depending on how many of these are already in place, feel free to skip to the step most relevant to you.

1.1.1 Obtain a server (either your own hardware or cloud provider)

A Linux Operating System (*OS*) is recommended for best ease of use and for taking maximal advantage of docker later. One such *OS* that VA Explorer has been reliably tested on is Ubuntu. To get a server:

- If using a cloud provider, select and provision one of their Ubuntu server instances according to their instructions.
- If provisioning your own server, you may be able to do so through your organization or hosting provider (see Step 2).

While not required, hardware specifications for the server that has successfully hosted VA Explorer in the past had a reference baseline of 1 CPU, 16 GB RAM, and 1 TB Storage.

1.1.2 Obtain a web address/domain name

A domain name such as `my-va-explorer.com` is required for users to navigate to your server from a browser. If you already own a domain, a subdomain (`my-va-explorer.my-website.com`) is a good approach too. If you already own a domain name, please proceed to the next step.

For those wishing to acquire a domain name, your organization may be able to provide one or you can pay a commercial domain registrar for one. Please note that new domains can take some time to be accessible from the Internet. While waiting, you need to direct this domain to the hosted server from Step 1:

- If using a cloud provider, please follow their instructions on connecting domain names to their hosted server instances.
- If you've provisioned your own server through your organization or hosting provider but they are not able to provide a domain name, that server should have an IP address. You'll need to follow the instructions of your domain registrar to point the newly registered domain name to your server's IP address.

1.1.3 Prepare server to host VA Explorer

To bring your server from blank slate to ready-to-host, there are final some steps to take before downloading VA Explorer.

- Install Docker Engine on Ubuntu
- Install Docker Compose Standalone
- Post Installation Setup for Docker Users on Linux
- Install a web server/ reverse proxy to direct web traffic to the VA Explorer port. Two popular options are Nginx and Apache. An example Apache configuration is provided in the [Troubleshooting](#) (page 42) section
- If your hosting provider doesn't already or automatically provide https support, you should add it now. Let's Encrypt is a free https certificate provider that also provides a guides for adding https support to Apache or Nginx web servers, among other configurations.
- Tools like git and text editors like vim will also be needed. You can install them via `sudo apt install -y git vim` or your OS's equivalent

1.1.4 Deploy and run VA Explorer via Docker

VA Explorer itself is quick to deploy thanks to Docker and the default configurations. More on customizing this configuration can be found in [Configuration & Deployment](#) (page 6).

- Assuming you've ssh'd into the server, navigate to the directory you'd like to install VA Explorer in.
- Retrieve the latest version of VA Explorer via: `git clone https://github.com/VA-Explorer/va_explorer.git` then navigate into the created directory via `cd va_explorer`
- You can provide initial configuration to supplement defaults by creating a `.env` file. Do so via `mv .env.template .env` and `vim .env`. One variable that can be edited immediately is the `DJANGO_ALLOWED_HOSTS` variable by adding your domain name from Step 2. For more information on this, see the mentioned [Configuration & Deployment](#) (page 6) section.
- Run `docker-compose up -d --build`. If you experience any issues here, please consult the [Troubleshooting](#) (page 42) section.
- Additionally, the full and complete list of management commands is available by running `manage.py help`. Only some of the popular commands are described here or in [Management Commands](#) (page 36).

- VA Explorer should now be up and running. You can confirm services are running as expected via `docker ps` which should output like below:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
172bb6c67e6b	va_explorer/pycrossva	"/entrypoint.sh /sta..."	3 hours ago	Up 3 hours	443/tcp, 0.0.0.0:5001->80/tcp, :::5001->80/tcp	va_explorer_pycrossva_1
2710b8f4bddb	va_explorer/celeryworker	"/entrypoint /start..."	6 weeks ago	Up 3 hours		va_explorer_celeryworker_1
69bec32648a9	va_explorer/django	"/entrypoint /start..."	6 weeks ago	Up 3 hours	0.0.0.0:5000->5000/tcp, :::5000->5000/tcp	va_explorer_django_1
ba5e731d1e7b	va_explorer/flower	"/entrypoint /start..."	6 weeks ago	Up 3 hours		va_explorer_flower_1
fe43f8dec370	va_explorer/interva5	"Rscript ../api.R"	6 weeks ago	Up 3 hours	0.0.0.0:5002->5002/tcp, :::5002->5002/tcp, 8787/tcp	va_explorer_interva5_1
b87a6306d208	redis:5.0	"docker-entrypoint.s..."	6 weeks ago	Up 3 hours	6379/tcp	va_explorer_redis_1
f50e0ea95273	va_explorer/postgres	"docker-entrypoint.s..."	6 weeks ago	Up 3 hours	5432/tcp	va_explorer_vapostgres_1

Fig. 1: `docker ps` output showing the pycrossva, clereryworker, django, flower, interval5, redis, and postgres services running

1.1.5 Create your first Admin account and log in

Finally, to get started in the application you should create an admin account for yourself. This can be done via built-in management command from within the main web application container. To seed an admin user for yourself:

- Open a shell within the main web application docker container by running `docker exec -it va_explorer_django_1 bash`
- Run the `seed_admin_user` command via `manage.py seed_admin_user <EMAIL_ADDRESS>` filling in your own email for the email parameter
- A temporary password will print to the console. Copy it. In a browser, navigate to the domain name you chose in Step 2 and log in with the email you provided plus the password you copied.
- You should now be logged in and immediately prompted to reset your password to something of your choosing. Take the opportunity to do so now.

Once you have one account you do not have to repeat this process as the application provides admins an interface for creating and managing other users.

With this basic setup in place you are ready to get started. As mentioned, if you are interested in reading more about basic or advanced configuration options like allowing an email server to handle user password resets you can find that in the next section [Configuration & Deployment](#) (page 6). You may also want to read over [IT Support Guides](#) (page 39) for information on activities like setting up an email server, implementing server monitoring, and preparing for regular backups.

1.2 Configuration & Deployment

VA Explorer configuration is primarily set by `docker-compose.yml` with sensible defaults. Admins or *IT* Staff with access to the server hosting a VA Explorer instance can change this configuration via the `.env` file. To begin using this file (if you haven't already during Installation & Setup's Step 4) you can use `mv .env.template .env` and `vim .env` to open a `.env` file for editing. To give an idea for how the VA Explorer could be changed each of the options is documented below.

Table 1: List of all configuration options for the base VA Explorer app along with default values and descriptions for each, plus external references where appropriate

Variable Name	Default Value	Description
DJANGO_DEBUG	Not Set	True or False. Will configure VA Explorer to report additional info in logs, error pages, console messages, and via a debug toolbar if True. Not set by default to allow settings.local or settings.production to handle value.
EMAIL_URL	consolemail://	A Django Environ ¹ email_url which supports various email related URL schemas. Used to allow VA Explorer to connect to an email server and send email on your behalf. Setup to print to the console (docker logs) by default. Recommended to customize.
DJANGO_DEFAULT_FROM_EMAIL	VA Explorer <noreply@vaexplorer.org>	The email address used in the sender field when VA Explorer sends automated emails. Format options include Name <email> or email. Default is app value here instead of Django default ² . Recommended to customize.
DJANGO_SECRET_KEY	dcc02e52ccbb649b9feb e9182abfa5e03c49be6c	Hash used by Django to cryptographically sign ³ things like sessions and account recovery email urls. Defaults to the given hard-coded random hash. Recommended to customize.
DJANGO_ALLOWED_HOSTS	localhost	A list of comma separated ip:port or unix:path formatted strings representing the host's or domain names that VA Explorer can serve. A security measure to prevent Host header attacks. Defaults to local computer browser only. Recommended to customize.
CELERY_BROKER_URL	redis://redis:6379/0	A Celery supported backend ⁴ URL schema. Allows VA Explorer to support reading results from long-running tasks such as batch cause-of-death assignment. Points to built-in redis service by default.
REDIS_URL	redis://redis:6379/0	ip:port or unix:path format location for Django caching solution ⁵ . Default allows VA Explorer to take advantage of same redis available as celery backend so is often the same value.

¹ <https://django-envIRON.readthedocs.io/en/latest/types.html#environ-env-email-url>

² https://docs.djangoproject.com/en/4.1/ref/settings/#std-setting-DEFAULT_FROM_EMAIL

³ <https://docs.djangoproject.com/en/4.1/topics/signing/>

⁴ <https://docs.celeryq.dev/en/stable/getting-started/backends-and-brokers/index.html>

⁵ <https://docs.djangoproject.com/en/4.1/topics/cache/>

POSTGRES_HOST	vapostgres	Value indicating the postgres host location. Formatted as a PostgreSQL host ⁶ parameter. Default value points to built-in postgres docker container.
POSTGRES_PORT	5432	Value indicating the port postgres runs on at the host location. Formatted as a PostgreSQL port ⁷ parameter. Default points to postgres port of built-in service.
POSTGRES_DB	va_explorer	Value indicating the name of the database used by VA Explorer. Formatted as a PostgreSQL db-name ⁸ parameter. Default is name of app.
POSTGRES_USER	postgres	Value indicating the name of the user accessing postgres. Formatted as a PostgreSQL user ⁹ parameter. Defaults to standard user.
POSTGRES_PASSWORD	postgres	Value indicating the password to use for the user accessing postgres. Formatted as a PostgreSQL password ¹⁰ parameter. Defaults to standard pass of built-in service. Recommended to customize.
QUESTIONS_TO_AUTODETECT_DUPLICATES	"Id10017, Id10018, Id10019, Id10020, Id10021, Id10022, Id10023"	A list of comma separated fields corresponding to questions on a VA (See standard) ¹¹ . Allows for customization of which fields VA Explorer considers when attempting to detect duplicate VAs. Defaults to fields having to do with name, sex, date of birth, and date of death.

Config values are read from `.env` first, then `docker-compose.yml` if unset, and finally from framework settings in `config/settings/production.py` during end-user docker builds when config hasn't been set elsewhere.

Note: If you update any of these configuration variables, please also run `docker-compose up -d` once more to push your `.env` file updates to the various containers.

For further configuration information, particularly for integrating with external services such as [ODK](#) and [DHIS2](#), please see [Integrations](#) (page 17).

1.3 Tips & Best Practices

As a compliment to your new VA Explorer instance, there are several optional steps you can also take to make server management easier and data more resilient. This section covers some of these additional steps as a good starting set:

- Setting up a mail server
- Configuring for better security
- Implementing server monitoring

There are also further guides along these lines in [IT Support Guides](#) (page 39) for those anxious to learn more about sustaining use of VA Explorer over time.

⁶ <https://www.postgresql.org/docs/current/libpq-connect.html#LIBPQ-CONNECT-HOST>

⁷ <https://www.postgresql.org/docs/current/libpq-connect.html#LIBPQ-CONNECT-PORT>

⁸ <https://www.postgresql.org/docs/current/libpq-connect.html#LIBPQ-CONNECT-DBNAME>

⁹ <https://www.postgresql.org/docs/current/libpq-connect.html#LIBPQ-CONNECT-USER>

¹⁰ <https://www.postgresql.org/docs/current/libpq-connect.html#LIBPQ-CONNECT-PASSWORD>

¹¹ <https://www.who.int/standards/classifications/other-classifications/verbal-autopsy-standards-ascertaining-and-attributing-ca>

1.3.1 Setting up a mail server

If a mail server is already available, customize the `EMAIL_URL` configuration variable to point to it with the necessary authentication information. However, if your organization or cloud provider do not provide a mail server, it is recommended to set one up. A popular and readily compatible tool for this is EXIM. Ubuntu provides documentation on installing and setting EXIM up locally. Once this process has finished, `EMAIL_URL` can similarly be customized to point to `localhost`.

Note: If users are reporting non-receipt of emails, the *Frequently Asked Questions* (page 42) section may help.

1.3.2 Implementing server monitoring

After installation and setup of VA Explorer, having visibility of system statistics or alerts when things do not function as expected can help you quickly address problems before they grow. If your organization or cloud provider offers this functionality, consider following their instructions to implement monitoring. If they do not already provide a way to do this, NetData OSS Agent may meet your needs as a nice tool providing out-of-the-box monitoring and visualization with low-to-no configuration. With some configuration, alerts are also supported.

NetData and other monitoring solutions should cover metrics that provide insight for common VA Explorer management questions like:

- Are backups, *VAs*, etc. overflowing storage? – Monitor Disk Space Utilization
- Is software running efficiently? – Monitor CPU & RAM Usage
- How well is the network/ server handling user traffic? Monitor Network Traffic/ Latency
- Is VA Explorer accessible? – Monitor Endpoints (Health Checks/ Pings)

1.3.3 Configuring things for better security

While VA Explorer is developed with security in mind and server software may have a good security baseline, security is a broad and continuously evolving domain. No set of things can fully reduce risk in this area, but some security best practices to consider for your VA Explorer server include:

Strengthening server authentication methods

If your server is using password authentication, ensure they are strong passwords and consider enabling 2-Factor authentication. Also consider using ssh key pairs instead of passwords.

Regularly updating server software

Software that ships with your server, including VA Explorer, periodically release updates, and sometimes these updates have the sole purpose of patching security vulnerabilities. Failing to download these security updates leaves software open to known vulnerabilities. To update server software, regularly run `sudo apt update && sudo apt upgrade` or your *OS* equivalent. To update VA Explorer, refer to the *IT Support Guides > Upgrading VA Explorer* (page 40) guide.

You may also consider setting your server up for automatic security updates. Debian-based Linux systems like Ubuntu provide the `unattended-upgrade` utility for this, along with optional configuration to alert you if any problems arise during automatic updates. See [Guide¹²](#).

¹² <https://help.ubuntu.com/community/AutomaticSecurityUpdates>

Protecting against server attacks

Servers can come under attack within minutes of going online to the wider internet. If you do not have an organization or cloud provider already protecting your server, some small tweaks that can help with this include the following. Note that this list is by no means exhaustive, and conducting a full cybersecurity review is recommended.

- Reducing your “attack surface” by closing or blocking any open but unneeded ports. You can do so through tools like `ufw`, `netstat`, `chkconfig`, and others.
- Consider switching popular services to non-standard ports. For example, `ssh` is usually supported via port 22, and expected to be found there because of that. Hosting it on another port could be a simple but effective method at reducing attacks.
- Consider removing the ability to directly login as root and instead make other users become root once logged in. This can be done by editing `/etc/passwd` root config to `root:x:0:0:root:/root:/sbin/nologin`
- Finally, tools like Fail2Ban, an intrusion prevention framework, can monitor for brute force attempts on services like `ssh`, Apache/ Nginx logs, etc. and then block offending IP addresses from future attempts. Info on taking advantage of its features is readily available from the community.

FEATURES

VA Explorer is an open-source web application built to help individuals and organizations manage, analyze, and disseminate *VAs* while integrating with existing *CRVS* systems at scale, simplifying the use of *VAs* for those individuals and organizations. To that end, VA Explorer offers some major features:

- Tablet Friendly Interface
- User Identity & Access Management
- VA Data Import
- Data Collection Supervision
- Processing & Analyzing Cause of Death Data
- Repairing VA Errors
- VA Search & Retrieval
- Exporting Data

VA Explorer is in active development. A prototype, with the features described here and screenshot with fake data, has been piloted in Zambia and is expected to continue improving in response to new research and user feedback. If you would like to see something specific from VA Explorer going forward, we look forward to hearing from you! Please submit feedback, bug reports, feature requests, etc. to our open-source project page.

For information on who should use VA Explorer, this documentation covers *User Training* (page 30) for all user and support roles. Information on potential levels of support that may be required to run a tool like VA Explorer is also available via our *IT Support Guides* (page 39) training series.

2.1 Tablet Friendly Interface

VA Explorer needs to support users on desktops and mobile devices. This support enables those whose roles place them in an office as well as those whose roles take them out to fieldwork. To that end, VA Explorer's interfaces are designed to display comfortably at iPad resolution minimums (1024px x 768px) and scale easily to larger devices.

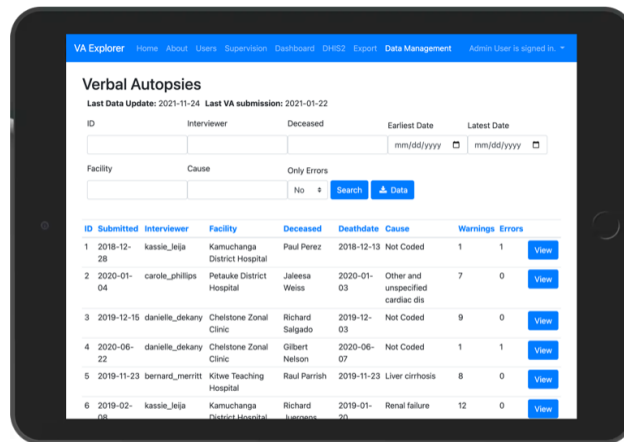


Fig. 1: Picture of VA Explorer, comfortably visible on an iPad

2.2 User Identity & Access Management

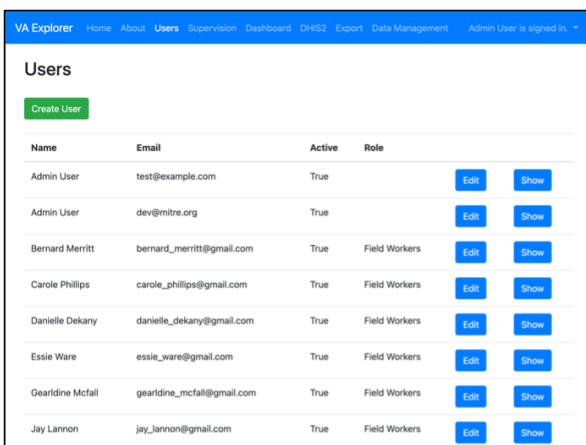


Fig. 2: Users module: A list of users

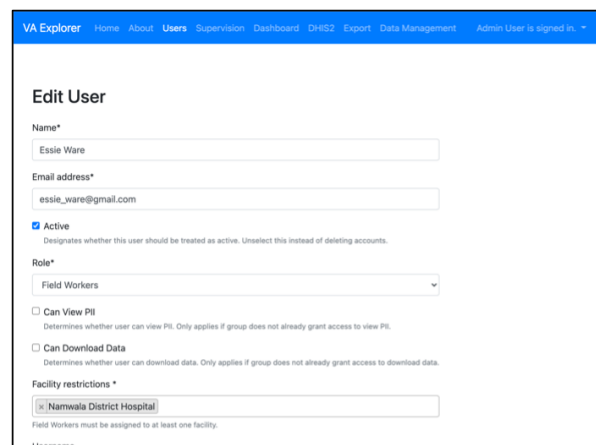


Fig. 3: Users module: An individual user edit form

VA Explorer supports different user types and levels of access for those users to facilitate roles-based [VA](#) workflows. This means VA Explorer offers features like:

- Creation & management of user accounts by administrators
- Ability to disable, restrict access to data by feature, role assignment, or geography
- Ability for users to manage their own passwords automatically

2.3 VA Data Import



Fig. 4: Homepage charts showing imported VAs + statuses over time

ID	Submitted	Interviewer	Facility	Deceased	Deathdate	Cause	Warnings	Errors
1	2022-04-30	nosaj_thomas	Mazabuka General Hospital	Richard Roe	2022-04-30	Not Coded	0	0
2	2022-08-04	nosaj_thomas	Chipata Central Hospital	Richard Roe	2022-08-04	Liver cirrhosis	7	0
3	2022-08-04	nosaj_thomas	Chilanga Health Centre	Richard Roe	2022-08-04	Not Coded	1	0
4	2022-05-06	nosaj_thomas	Mumbwa District Hospital	Jane Doe	2022-05-06	Not Coded	1	0
5	2022-08-04	nosaj_thomas	Chawama Level One Hospital	Richard Roe	2022-08-04	Liver cirrhosis	7	0
6	2022-05-07	nosaj_thomas	Matero Level One Hospital	Jane Doe	2022-05-07	Not Coded	1	0

Fig. 5: Users module: An individual user edit form

VA Explorer has limited functionality on its own – the true value of the tool comes from how it helps you process your VA data. To obtain this data, VA Explorer currently supports different data import methods (enumerated below). After import, all VAs are searchable/ filterable via a number of useful parameters such as interviewer, name of deceased, date, facility, cause of death, and whether the VA has errors.

- **(Automatic) Click Import Data Button:** This format is the most user friendly: the VA Explorer UI will detect whether you are using [ODK](#) or a more general [CSV](#) import and react accordingly.
- **(Manual) Import from CSV:** This format allows the most flexibility: if your VAs can be exported from their origin in [CSV](#) form, you can import them into VA Explorer. See [Admin Guides > Data Admin](#) (page 36) training for specifics on how.
- **(Manual) Import from ODK:** This format implements manual support for [ODK](#) integration: bring your VAs directly over from [ODK](#) like clicking the data import button, but with more customization about how and when. See [Admin Guides > Data Admin](#) (page 36) training for specifics on how.

2.4 Data Collection Supervision

VA Explorer allows admins and data managers to supervise the origins of collected VAs. Want to know which facilities are generating the most VA errors when attempting to assign a [CoD](#)? Want to know which interviewers/ field workers are collecting the most VAs? This feature helps supervisor roles answer those types of questions about the overall VA dataset.

- Overview of total VAs, total warnings/errors thrown during the [CoD](#) assignment, and performance stats such as VAs/week
- Filter VAs under supervision by dates of interest
- Group data by interviewers/field workers or by facility for a different set of insights

VA Explorer: Home About Users Supervision Dashboard 2022 Export Data Management Data Cleaner Admin User is signed in

Facility Supervision

Group by: Facility
 Facility: Earliest Date: Latest Date:

Facility	Total VAs	Warnings	Errors	Weeks of Data	Last Submission	VAs / week
Petauke District Hospital	31	113	6	16	2022-10-19	1.94
Macha Mission Hospital	30	136	5	18	2022-10-20	1.67
Chilanga Health Centre	30	62	9	15	2022-10-19	2.0
Ndola Teaching Hospital	27	83	7	15	2022-10-15	1.8
Senanga General Hospital	25	85	5	15	2022-10-19	1.67
Namwala District Hospital	24	71	4	13	2022-10-20	1.85
Nchanga North General Hospital	23	98	4	16	2022-10-11	1.44
Mumbwa District Hospital	23	82	3	14	2022-10-16	1.64
Kitwe Teaching Hospital	23	81	2	16	2022-10-20	1.44
Mwendere Mission Hospital	21	71	4	12	2022-10-19	1.75
Kalulushi General Hospital	21	81	4	13	2022-10-20	1.62
Livingstone Central Hospital	21	78	4	12	2022-10-19	1.75
Chongwe District Hospital	21	82	3	14	2022-10-16	1.5
Chipata Level One Hospital	21	65	2	13	2022-10-13	1.62
Njasa Clinic	20	74	8	14	2022-10-19	1.43
Rosan Antelope General Hospital	20	63	6	11	2022-10-07	1.82
Ronald Ross Hospital	19	38	10	12	2022-10-15	1.58
University Teaching Hospital	19	46	5	12	2022-10-20	1.58
Mtshwane Mission Hospital	19	39	4	13	2022-10-20	1.4

Fig. 6: Supervision module: VA stats grouped by facility

2.5 Processing & Analyzing of Cause of Death Data

Verbal Autopsy Actions

VAs With Coding Issues

ID	Submitted	Interviewer	Facility	Deceased	Deathdate	Cause	Warnings	Errors	
1	2022-04-30	nosa_j_thomas	Mazabuka General Hospital	Richard Roe	2022-04-30		0	0	<input type="button" value="View"/>
4	2022-05-06	nosa_j_thomas	Mumbwa District Hospital	Jane Doe	2022-05-06		1	0	<input type="button" value="View"/>
392	2022-08-31	nosa_j_thomas	Kalomo Hospital	Richard Roe	2022-08-31		1	1	<input type="button" value="View"/>
6	2022-05-07	nosa_j_thomas	Matero Level One Hospital	Jane Doe	2022-05-07		1	0	<input type="button" value="View"/>
7	2022-05-07	nosa_j_thomas	Lewanika General Hospital	Jane Doe	2022-05-07		1	0	<input type="button" value="View"/>

There are 529 additional issues not listed here

VAs With Indeterminate Cause of Death

ID	Submitted	Interviewer	Facility	Deceased	Deathdate	Cause	Warnings	Errors	
601	2022-07-17	nosa_j_thomas	Kalulushi General Hospital	Jane Doe	2022-07-17	Indeterminate	10	0	<input type="button" value="View"/>

Fig. 7: Homepage charts showing imported VAs + statuses over time

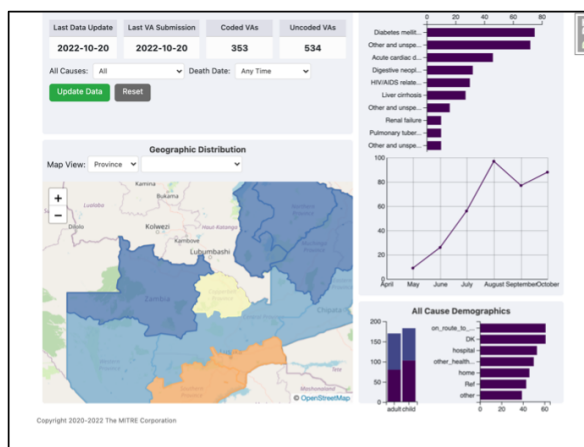


Fig. 8: Users module: An individual user edit form

After importing VA data, one of VA Explorer’s key benefits is the ability to quickly assign a *CoD* to each by running InterVA5. To enable this, VA Explorer currently supports two ways to process VAs via *CoD* assignment, enumerated below. The home page lists a quick reference of VAs that either contain coding errors to be addressed or were assigned “Indeterminate” as the cause of death and may require further investigation.

- **(Automatic) Click Run Coding Algorithms Button:** This format is the most user friendly: the VA Explorer UI will detect all uncoded VAs and attempt to run cause of death assignment on each or report any warnings/errors associated with VAs that cannot receive *CoD* assignment.
- **(Manual) Run Coding Algorithms:** This format implements manual support for InterVA5 integration: directly control the *CoD* assignment process for your VAs via management command. See [Admin Guides > Data Admin](#) (page 36) training for specifics on how.

Additionally, VA Explorer provides a dashboard of summary information, analytics, and charts for VAs that have successfully assigned a cause of death. The dashboard currently supports

- Showing an overview of VA processing progress with most recent dates and number of coded compared to uncoded VAs

- Understanding geographical trends via a dynamic heat map, with zoom capabilities to filter for regions of interest
- Showing cause of death plots for chosen regions
- Showing death distributions by age, gender, and place of death for chosen regions
- Showing trends over time for chosen regions
- Ability to filter dashboard data by cause of death, dates of interest, and geography

2.6 Working with VA Questionnaires



Field Name	Field Question	Value
username	Username	field_worker_59
submissiondate	Submission Date	2022-08-31 22:31:59.837178+00:00
Id10002	Is this a region of high HIV/AIDS mortality?	low
Id10003	Is this a region of high malaria mortality?	high
Id10004	During which season did (s)he die?	wet
Id10007	What is the name of VA respondent?	Jason Thomas
Id10008	What is your/the respondent's relationship to the deceased?	child
Id10009	Did you/the respondent live with the deceased in the period leading to her/his death?	no
Id10010	Name of VA interviewer	nosaj_thomas
Id10013	Did the respondent give consent?	yes
Id10017	What was the first or given name(s) of the deceased?	Richard
Id10018	What was the surname (or family name) of the deceased?	Roe
Id10019	What was the sex of the deceased?	male

Fig. 9: Record view showing VA responses.



Verbal Autopsy #596

Record Coding Issues Change History

5 Algorithm Warnings

- W610120-a (ll <3w) value inconsistent with W610120-b (ll 3+w) - cleared in working information 229 W610194-o (abd pain) not flagged in category W610193-o (abdom) - updated in working information
- W610157-o (cou bid) not flagged in category W610153-o (cough) - updated in working information 231 W610194-o (abd pain) not flagged in category W610193-o (abdom) - updated in working information
- W610176-a (chp 3d) not flagged in category W610174-o (ch pain) - updated in working information 234 W610194-o (abd pain) not flagged in category W610193-o (abdom) - updated in working information
- W610186-o (drr bid) not flagged in category W610181-o (diarr) - updated in working information 239 W610194-o (abd pain) not flagged in category W610193-o (abdom) - updated in working information
- W610263-b (swa liq) not flagged in category W610261-o (swallow) - updated in working information 246 W610194-o (abd pain) not flagged in category W610193-o (abdom) - updated in working information

Copyright 2020-2022 The MITRE Corporation

Fig. 10: VA Issues tab, showing algorithm errors preventing coding

Ideally the [VAs](#) collected by you, or your organization, are free of error. But for occasions where they are not, VA Explorer supports troubleshooting and correcting individual [VA](#) warnings or errors preventing cause of death assignment, or those leading to an “Indeterminate” assignment by InterVA5.

- Field workers are able to view and edit [VAs](#) for their location
- Data Managers and Admins can do repairs for any [VA](#) they have access to
- View errors and warnings causing issues with [VA](#) from both VA Explorer and InterVA5
- Easily edit [VA](#) answers with [VA](#) instrument compliant responses
- View change history and revert changes as needed for a [VA](#)

2.7 Processed Data Export

Finally, when users would like to send the [VAs](#) processed by VA Explorer onto another step in their analysis, or just save a copy for themselves, VA Explorer supports data export in both [CSV](#) and [JSON](#) formats. Additionally, if VA Explorer has been configured to integrate with [DHIS2](#) then users can export their data directly to that service. See [DHIS2](#) (page 19) for more info.

- Choose between [CSV](#) and [JSON](#) data download
- Filter data downloaded to just the [VAs](#) of interest
- Optionally export direct to [DHIS2](#) if your configuration supports it

Download Data ▼

Either export data to an external database (i.e. DHIS2) or download locally

Earliest Date
mm/dd/yyyy

Earliest VA date to download

Latest Date
mm/dd/yyyy

Latest VA date to download

Locations
Search for location(s) from which to download data
Choose location(s) from which to download data

Causes
Filter data by Cause of Death (CoD)
Filter data by Cause of Death (CoD)

Data Format
CSV ▼

Download

Copyright 2020-2022 The MITRE Corporation

D.D.T.

Fig. 11: Export module: options form for data export

INTEGRATIONS

VA Explorer supports integrations with a variety of services, some built-in, some external, in order to facilitate the quick and efficient movement or processing of [VAs](#). In this section you will find information on these integrations as well as details on how to enable them in VA Explorer if needed.

- ODK Central
- KoboToolbox
- DHIS2
- Algorithm Support
 - pyCrossVA
 - InterVA5

3.1 Networking

When integrating services with VA Explorer, operators should be aware of additional networking configuration if multiple docker-compose services are run on the same server.

docker-compose creates all containers that are part of its compose file under a default bridge network to allow inter-communication for that service. This setup does not allow one set of docker-compose managed containers to talk to another (VA Explorer -> ODK for example) by default.

To support this use case, operators can create an external bridge network and connect it to the necessary containers like so:

```
docker network create <external network name>
docker network connect <external network name> va_explorer_django_1
docker network connect <external network name> <other service's container name>
```

VA Explorer provides some additional network configuration to enable use of this external network via `.env` variables:

Table 1: List of all configuration options when integrating with ODK Central, plus default values and descriptions for each

Variable Name	Default Value	Description
USE_GATEWAY	False	True or False. Used by VA Explorer determine if an external docker network should be used by integrations. Defaults to <i>False</i> .
DOCKER_GATEWAY	""	An http scheme (<code>http://</code> or <code>https://</code>) followed by the IP address given by <code>docker network inspect <external network name></code> command. Found as the <code>IPAM.Config.Gateway</code> property's value. Used by VA Explorer as an adaptor for <code>HOST</code> value of the service you wish to integrate with. Defaults to an empty string.

These `.env` variables should supplement the variables described below in whichever service(s) you wish to integrate with.

3.2 ODK Central

VA Explorer supports integration with *ODK* Central to provide both manual and automatic import of *VA* questionnaire responses. You can read more about *ODK* Central via [its documentation](#)¹³.

- To perform manual imports of *ODK* Central data, please see *Data Admin* (page 36) in the Admin training guides.
- To configure VA Explorer for automatic import of *VAs*, the relevant `.env` variables are detailed below. Once properly set, run `docker-compose up -d` to push the new configuration. *ODK* integration features within VA Explorer should now appear and function.

Table 2: List of all configuration options when integrating with ODK Central, plus default values and descriptions for each

Variable Name	Default Value	Description
ODK_HOST	""	<code>ip:port</code> or <code>unix:path</code> format location for the ODK Central instance to connect to. Defaults to an empty string for no connection. Recommended to customize.
ODK_SSL_VERIFY	True	True or False. Used by VA Explorer determine enforcement of valid ssl/https connections. Defaults to <i>True</i> for enforcement.
ODK_PROJECT_ID	2	Value indicating the ID of the project in ODK Central that holds Verbal Autopsy forms or data. Typically, a number found in the url when viewing the project in ODK. Defaults to 2 (ex. <code>./#/projects/2</code>)
ODK_FORM_ID	va_who_v1_5_3	Value indicating the ID of the Verbal Autopsy form within the given project. Typically, found in the form list under the "ID and Version" column. Defaults to the ID of an example VA form.
ODK_EMAIL	user@example.com	Value indicating the email of the account you wish to use to login to ODK Central. Defaults to an example email. Recommended to customize.
ODK_PASSWORD	""	Value indicating the password for the provided email's account. Defaults to a blank string. Recommended to customize.

¹³ <https://docs.getodk.org/central-intro/>

3.3 KoboToolbox

VA Explorer supports integration with KoboToolbox as an alternative to ODK that also allows for both manual and automatic import of [VA](#) questionnaire responses. You can read more about KoboToolbox via [its documentation](#)¹⁴.

- To perform manual imports of KoboToolbox data, please see [Data Admin](#) (page 36) in the Admin training guides.
- To configure VA Explorer for automatic import of [VAs](#), the relevant `.env` variables are detailed below. Once properly set, run `docker-compose up -d` to push the new configuration. Kobo integration features within VA Explorer should now appear and function.

Table 3: List of all configuration options when integrating with KoboToolbox, plus default values and descriptions for each

Variable Name	Default Value	Description
KOBO_HOST	""	ip:port or url format location for the KoboToolbox instance to connect to. Defaults to an empty string for no connection. Recommended to customize.
KOBO_SSL_VERIFY	True	True or False. Used by VA Explorer determine enforcement of valid ssl/https connections. Defaults to <i>True</i> for enforcement.
KOBO_API_TOKEN	""	Value used to authenticate requests to the KoboToolbox API. A string of ~40 lowercase letters/numbers. Found for your user at <code>KOBO_HOST/token</code> when logged in to the instance. Inherits your same permissions. Defaults to an empty string. Recommended to customize.
KOBO_ASSET_ID	""	Value identifying the project used to collect Verbal Autopsies within KoboToolbox. Typically found in the url when viewing the project on the instance (Ex. <code>{KOBO_HOST}/#/forms/uk8r5yolfuacxkjibsj7nw/summary</code>) Defaults to an empty string. Recommended to customize.

3.4 DHIS2

VA Explorer supports integration with [DHIS2](#) to export [VAs](#) that have been processed and assigned causes of death along for potential further use. You can read more about [DHIS2](#) via their documentation. Details on configuring [DHIS2](#) for use with Verbal Autopsies metadata is also available via the [DHIS2 VA program open-source project](#)¹⁵.

To configure VA Explorer for automatic export of [VAs](#), the relevant `.env` variables are detailed below. Once properly set, run `docker-compose up -d` to push the new configuration. [DHIS2](#) integration features within VA Explorer should now appear and function.

¹⁴ <https://support.kobotoolbox.org/>

¹⁵ https://github.com/verbal-autopsy-software/DHIS2_VA_program

Table 4: List of all configuration options when integrating with DHIS2, plus default values and descriptions for each

Variable Name	Default Value	Description
DHIS_HOST	""	ip:port or unix:path format location for the DHIS2 instance to connect to. Defaults to an empty string for no connection. Recommended to customize.
DHIS_SSL_VERIFY	True	True or False. Used by VA Explorer determine enforcement of valid ssl/https connections. Defaults to True for enforcement.
DHIS_ORGUNIT	wEVB21sQaHu	Value indicating the root Organization Unit UID within your DHIS2 organizational hierarchy. Should be available using the DHIS2 API Query: /api/organizationUnits?level=1. Recommended to customize.
DHIS_USER	admin	Value indicating the username of the account you wish to use to login to DHIS2 Defaults to DHIS2's existing default username credential. Recommended to customize.
DHIS_PASS	district	Value indicating the password for the provided username's account. Defaults to the default credentials for DHIS2's default admin account. Recommended to customize.

If you encounter any issues during integration, please reference the [Troubleshooting](#) (page 42) section.

3.5 Algorithm Support

VA Explorer provides built-in support for cause of death assignment via the InterVA5 computer coded VA ([CCVA](#)) algorithm without additional configuration. If you would like to change how these services are configured (including the HIV or Malaria prevalence variables for InterVA5) or point to a custom service instead, this section will help with that. As a future expansion, VA Explorer may provide built-in support for additional [CCVA](#) algorithms.

3.5.1 pyCrossVA

pyCrossVA is provided as a built-in docker service to allow VA Explorer to transform [VAs](#) into input suitable for cause of death assignment algorithms. You can read more about it via its project page.

Table 5: List of all configuration options when interfacing with the py-CrossVA service, plus default values and descriptions for each

Variable Name	Default Value	Description
PYCROSS_HOST	http://pycrossva:80	ip:port or unix:path format location for the py-CrossVA service used by VA Explorer to prepare VAs for algorithm input. Defaults to built-in pyCrossVA docker service.

3.5.2 InterVA5

InterVA5 is provided as a built-in docker service to allow VA Explorer VAs to be processed for likely cause of death. You can read more about it via its project page.

Table 6: List of all configuration options when interfacing with the InterVA service, plus default values and descriptions for each

Variable Name	Default Value	Description
INTERVA_HOST	http://interva5:5002	ip:port or unix:path format location for the InterVA service used by VA Explorer to assign causes of death. Defaults to built-in InterVA docker service.
INTERVA_MALARIA	l	One of the enumerations “h” (high), “l” (low), or “v” (very low). Used to indicate the prevalence of Malaria within the population. Defaults to “l” for low.
INTERVA_HIV	v	One of the enumerations “h” (high), “l” (low), or “v” (very low). Used to indicate the prevalence of HIV within the population. Defaults to “v” for very low.
INTERVA_GROUPCODE	False	True or False. Used to set whether the InterVA cause grouping code is included with cause of death output. Defaults to False

Part II

Guides

GENERAL INFORMATION

Welcome to VA Explorer, the open-source application that helps you or your organization manage and analyze verbal autopsies. This section will help you, as an end user, become familiar with VA Explorer through three simple steps.

1. **Navigate to your VA Explorer instance**

Where did you or your admin setup VA Explorer on the web? Go there in your browser and sign in or request an account from your admin as needed.

2. **Explore features based on your role**

Read through our [Features](#) (page 11) section to get an idea what VA Explorer offers. You should have a role from when you created your account or from when one was created for you. See the [User Guides](#) (page 30) here for role-specific guidance.

3. **Practice taking supported actions**

Whatever your role, you have the power to take action in VA Explorer! Try some of the actions out as you learn about them in this training, and you will build a solid foundation for working with [VAs](#) in VA Explorer going forward.

Outside of these three steps (which reference this training), modules are organized into general guidance, role-specific user guides, admin guides, [IT](#) guides, and finally, frequently asked questions. Please feel free to submit feedback if you find anything missing after reading and know that the VA Explorer team wishes you all the best in your [VA](#) management experience!

4.1 The VA Questionnaire

VA Explorer's foundation centers around the management of Verbal Autopsies ([VAs](#)) which are conducted by people and organizations around the world according to the 2016 World Health Organization ([WHO](#)) verbal autopsy instrument. VA Explorer will support the 2022 [WHO](#) verbal autopsy instrument in the future, when the [CCVA](#) algorithms also support the 2022 questionnaire. Use of non-standard [VA](#) instruments may cause issues in VA Explorer if attempted.

4.2 Your VA Explorer Account

An administrator creates Accounts in VA Explorer. Your account role and any additional settings like geographic restrictions determine the parts of the VA Explorer you can access and the actions you can take. If you are an admin, this training covers this workflow.

If you are not an admin and you think your permissions are misconfigured, contact your system's admin.

4.2.1 User Permissions

There are four VA Explorer user roles, each with different capabilities. A user may only have one role per account/email address, but there may be multiple users assigned to each type of role (e.g., a user can only be one of Admin, Data Manager, Data Viewer, or Field Worker; but there may be multiple Admins, Data Managers, Data Viewers, and/or Field Workers). The table below provides a high-level description of each role.

Table 1: List of available roles and their descriptions

Role	Description
Admin	Admins oversee the VA Explorer system. By default, Admins have full access to the application. This includes data for all VAs as well access to all system functions such as managing accounts, importing, processing, and exporting data.
Data Manager	Data Managers have near full access VA Explorer, equivalent to admin access without the ability to create and manage user accounts. Has full access to VA data and system functions unless any restrictions (ex. Geographic) are placed on them by the admin.
Data Viewer	Data Viewers have access that allows them use VA Explorer for reporting purposes – a Data Viewer can view, analyze via the dashboard, or search through VAs but cannot modify any VAs
Field Worker	Field Workers have similar permissions to Data Viewers, but are allowed only allowed full access to view and modify VAs for their own location. This allows them to assist data managers in repairing any cause of death errors.

Users should have a role with the fewest capabilities and least record access required to perform their job. For example, if a user only needs to be able to view [VA](#) data, a Data Manager role would not be appropriate.

Geographic Access

Your admin may also associate your account with specific geographic regions or facilities. This assignment is called “Geographic Access” in VA Explorer. If your account has this restriction, VA Explorer only shows you [VAs](#) from your assigned regions or facilities. The default geographic restriction is the broadest “National” access, which shows [VAs](#) from all regions. To support additional location restriction options beyond the default, Admins should follow the guide in [Loading Locations for Geographic Access Support](#) (page 34)

Abilities to Download and View PII Data

Personally Identifiable Information (*PII*) is a category of information that can be used to identify an individual. Some users may be further restricted from viewing or downloading certain data such as [PII](#) or, separately, from downloading data altogether for privacy or policy reasons. Users do not have these restrictions by default, so Admins will determine if individual users have this restriction. If users have the download data restriction, they will not be able to use the Export feature. If users are restricted from viewing [PII](#), the following fields will not appear in the application or in downloaded data. Instead, fields will automatically be replaced with **** redacted ****.

Table 2: List of VA fields redacted for users without PII viewing permissions

VA Field	Meaning/Information Protected
Id10007	Name of VA respondent
Id10017	First name of subject
Id10018	Surname of subject
Id10061	Name of subject's father
Id10062	Name of subject's mother
Id10070	Subject death registration number
Id10073	Subject national identification number

4.2.2 Common Actions Across User Roles

Signing in for the First Time

Signing into VA Explorer requires an email address and password. You will use a temporary password obtained from either system-generated email or your administrator when signing in for the first time. Contact your admin if you do not have a temporary password or cannot find it.

1. Navigate to VA Explorer in your web browser. Compatible web browsers include Google Chrome, Microsoft Edge, Mozilla Firefox, and Apple Safari.
2. Use your email address and your temporary password to sign in. You will be prompted to choose a new password on the next page. Follow the instructions to create a strong password that will help protect your VA Explorer account.
3. After signing in, the application will take you to the Home page. The Homepage shows trends of [VAs](#) collected and coded, as well as a snapshot of [VAs](#) with coding issues and [VAs](#) coded with Indeterminate cause of death. The scope of [VA](#) data shown on the Home page depends on your assigned role and geographic access.
 - **For Field Workers:** If you are assigned the Field Worker role, the Homepage shows trends and data on the [VAs](#) for your location.
 - **For All Other Roles:** For all other roles, data on the Home page are limited to the specific regions or facilities you can access.

Viewing Your Account Profile

Your profile shows the role you have been assigned, your geographic access, and actions you can take in VA Explorer, such as viewing [PII](#) and downloading data. These account settings are described in [User Roles and Capabilities](#) (page 30) To view your account profile:

1. Click your name in the upper right-hand corner of the navigation bar; a dropdown menu will open.
2. Click the “My Profile” option within the dropdown menu.

Please contact your administrator if you need to update your account settings or if something looks incorrect. Only individuals with administrator roles can update user accounts.

Changing Your Password

To change your password while signed into VA Explorer:

1. Click your name in the upper right-hand corner of the navigation bar; a dropdown menu will open.
2. Click the “Change Password” option within the dropdown menu.
3. Enter your current password.
4. Choose a new password, following instructions to enter a valid password twice.

If you do not know your current password, sign out of VA Explorer and click “Forgot Password?” You will be instructed to enter the e-mail address associated with your account, and VA Explorer will send an email with a link allowing you to reset it.

Using the Analytics Dashboard

All users can view the [VA Analytics Dashboard](#) to see information on the [VAs](#) (filtered to their level of permissions and access) that have successfully been assigned a cause of death. To do so, click “Dashboard” in the navigation bar to view it.

The [VA Analytics Dashboard](#) is a dynamic, visualization-based dashboard that helps you explore cause of death data. It has three global filters that simultaneously update all graphs, maps, and statistics found in the top left of the dashboard page and directly above the heatmap. The global filters include:

Table 3: Specific descriptions and usage info for select filters available via the VA Explorer dashboard

Filter	Description	How to Use
Death Date	View analytics for a specific date of death time frame	Choose the time frame you want to explore from the Time Period dropdown menu (ex. Within 1 Month, Within 1 Year, or Custom)
Cause of Death	View analytics for a particular cause of death	Choose a cause of death or a category of causes (ex. Infectious, NCD, etc.) from the Causes dropdown menu
Location	Filter analytics to specific location(s)	Choose specific geographic regions using the province/district selector and location dropdown or click a region on the map.

Viewing Dashboard Components

The Dashboard includes several visualizations, including:

- A dynamic heatmap showing geographical trends, with ability to filter for regions of interest. Found in the bottom left of the dashboard page.
- Cause of death plots. Found in the top right of the dashboard page.
- Death distributions by age, gender, and place of death (“demographics”). Found in the bottom right of the dashboard page.
- Cause of death trends over time. Found in the middle right of the dashboard page.

Searching for and Reviewing Specific VAs

All roles have some level of access to search for and review specific VAs. To do so, click “Data Management” in the navigation bar to view the Data Management page.

The Data Management page shows a paginated table of all the VAs in the system your account is eligible to see. For more information about your account and eligibility, see [Viewing Your Account Profile](#) (page 26). On the Data Management page, you can search or filter available VAs with the following parameters:

Table 4: Specific descriptions for each VA property available to filter with

Field	Description
ID	The unique numeric identifier assigned to the specific VA in VA Explorer.
Interviewed	The date the VA interview was conducted. If your account does not have PII access, you will see “Date Unknown”.
Facility	The facility where the VA was collected. If facility information is missing, you will see “Location Unknown”.
Deceased	The name of the deceased individual. If your account does not have PII access, you will see “Subject Unknown”.
Deathdate	The date of the individual’s death. If the VA is missing this information, you will see “Date Unknown”.
Cause	The cause of death assigned by coding algorithm. If the VA has not yet been coded, “Not Coded” will be displayed. A VA is “Not Coded” if the algorithm has not been run or if error(s) prevent the VA from being coded.
Warnings	The count of warnings generated by the coding process. An example of a warning you might see for InterVA5 is: “field ageInYears2, age was not provided or not a number.”
Errors	The count of errors generated by the coding process. An example of an error you may see for InterVA is: “Error in age indicator: Not Specified.”

In addition to searching and filtering, users may also have access to the following actions depending on their permissions:

- Sort VA table data by column value by clicking any of the column headers. Clicking again reverses the sort order of the column
- Download VA table data by clicking the “Export” button to be directed to the Export page with your search parameters automatically filled into the export form.
- View VA details by clicking the “View” button on any individual VA. See details below.

Viewing Details for Individual VAs

When viewing a specific VA, the resulting page may show a couple of tabs at the top below the VAs ID. Users with permission to change VAs will also see the option to repair the VA via an “Edit Record” button. See [Repairing VA Errors & Warnings](#) (page 32) for details on editing VAs. Quick navigation options to automatically scroll to the top or bottom of the VA responses is available via the floating action button in the bottom right.

Record

This tab is always visible and shows the *VA* questionnaire data, including:

- The question ID corresponding to the *WHO* standard instrument
- The text of question associate that pairs with each question ID
- The response or calculation for each question

By default, all empty fields are hidden so that only questions with answers are shown. Revealing these is possible by unchecking the “Hide Empty Fields” box in the top right of this tab.

Coding Issues

This tab will become visible if the individual *VA* has any warnings or errors associated with it.

These warnings and errors can be applied during data import or, more often, after processing the *VA* for cause of death.

- **(Errors)** these are issues that either VA Explorer or a coding algorithm has determined is severe enough to totally prevent *CoD* assignment. They will need to be corrected.
- **(User Warnings)** these are issues that should be addressed because they potentially block *CoD* assignment or the accuracy of the assigned *CoD* but are not quite as severe.
- **(Algorithm Warnings)** these are warnings specifically provided by the algorithm after assigning a *CoD* that indicate the assignment may not be accurate. Fixing these increases the accuracy of *CoD* assignment. This tab will be visible if the individual *VA* has been edited or updated in any way since its initial import.

Change History

This history provides for transparency, record integrity, and protection from misuse for *VAs* and will take the form of an audit trail, a table of all changes, with each row showing

- The date of the change
- The user who made the change
- The fields before and after the change

Additionally, this tab will contain two action buttons:

- **(Reset to Original)** To completely reset the *VA* to the state it was in at the time of original import. (Note that for this action, change history is still preserved)
- **(Revert Most Recent Change)** To erase or undo the change described at the top of the change history table (organized by most recent changes first)

Duplicate VAs

If your system is configured to automatically flag potential duplicate *VAs*, you may also see a yellow warning banner above these tabs. This banner alerts you that VA Explorer has flagged this *VA* as a possible duplicate. Please consult with an administrator or data manager to determine any actions to take for potential duplicate *VAs*.

USER ROLES

In addition to the training previously detailed for *Common Actions* (page 26), users may want to read training specific to their role. This section offers guidance for Data Managers, Data Viewers, and Field Workers. For Admins, please see the next section *Admin Guides* (page 34) which covers your more expansive abilities.

5.1 Data Managers

As mentioned, Data Managers have near full access VA Explorer, equivalent to admin access without the ability to create and manage user accounts. They should support the admin in VA Explorer operations to reduce burden on Admins, who can then focus on ensuring Data Managers have the users they need (such as Data Viewers and Field Workers) to be effective. These guides help cover those tasks Data Managers can undertake.

5.1.1 Viewing Users & Supervision Information

As a Data Manager, you have access to the profiles of all VA Explorer users, but do not have the ability to make changes to them, such as adding permissions. This ability may be useful in determining if users you supervise have the correct access, or facilities you supervise have the right amount of coverage through user geographic access. To view the list of Users, click “Users” in the top navigation bar. To view an individual user, click the “Show” button next to their name.

Next, to view VA Explorer’s supervision information click “Supervision” in the navigation bar to visit the Supervision page. This page shows VAs collected in the system grouped by Interviewer/ Field Worker or by Facility. In each case, the page lists the following information to help you answer various supervision questions such as:

- Which interviewer or facility is generating the most issues when we attempt to code VAs?
 - Total Warnings
 - Total Errors
- Which interviewers or facilities are providing the most data and how often?
 - VAs / week
 - * Total VAs
 - * Total Weeks of Data
 - Date of Last Interview

When viewing data grouped by Interviewer/ Field Worker, this information (each sub-bullet beneath the questions) is available for each individual Field Worker (ex. Total VAs collected by interviewer John Smith, total for . . . , etc.) and shows the Facility that Field Worker is assigned to. However, when viewing data grouped by Facility, the names of individual Interviewers/ Field Workers go away as the information becomes summarized at the Facility level (e.g., John

Doe and Jane Roe each collected 20 VAs for Example Clinic so Total VAs now shows Example Clinic – 40 total VAs). Changing this grouping is done by toggling the “Group by” dropdown in the top form and clicking “Go”

Additionally, it is possible to filter the supervision data in case you’d like to see it for specific facilities or for certain date ranges (How many VAs did John Doe collect last month?). To apply a filter, enter in your parameters to the form at the top and click the “Go” button.

5.1.2 Exporting Data

Click “Export”, either in the navigation bar, search interface, or the dashboard to navigate to the Export page. When exporting data from VA Explorer for local use:

1. Either customize the data you would like to export by using the available filters or confirm that the form has been auto-filled with the filters you were using when navigating here from the search interface or dashboard. By default, if no filters are applied, the system will attempt to export all the data you have permissions to access.
2. Choose the export format: *CSV* or *JSON*.
3. Click “Download.” A modal will appear explaining that your download is being prepared and compressed to a .zip file. For the download to complete, do not navigate away from the Export page. The modal will close automatically when the download has finished. Please note that large file sizes may take a while to download.

5.1.3 Running Data Cleanup Operations

VA Explorer can autodetect VAs that are possible duplicates. The system flags VAs that match across a set of fields and shows these possible duplicates under the “Data Cleanup” navigation bar item. The Data Cleanup functionality is available when your system has been configured to autodetect potential duplicate VAs. By default, this feature is turned off. If you do not see the Data Cleanup menu item and think you should, contact your system administrator for more information.

As an example of how this feature works, let’s say your system is configured to identify VAs as duplicates if they match interview date, name, gender, and age of the deceased person. When VA Explorer detects two (or more) VAs that match across these fields, it leaves the oldest one unmarked and flags all others as possible duplicates.

The set of fields used to identify duplicates is configurable, and your system administrator may set and update them.

To manage what is done with these VAs flagged as potential duplicates, you may delete them or edit the possible duplicate VAs to remove them from detection. See the table below for all actions you can take on the data cleanup page.

Table 1: List of actions available through the data cleanup UI and what they accomplish

Action	Description
Download All	Download all potential duplicate verbal autopsies to a CSV file for examination.
Delete All	Delete all potential duplicate verbal autopsies. The oldest VA (by created timestamp) amongst a set of matching VAs is designated as the non-duplicate and kept in the system.
View	View a single VA flagged as a possible duplicate.
Download Individual	Download an individual verbal autopsy and verbal autopsy(ies) flagged as its potential duplicate(s) to a CSV file for examination.
Delete Individual	Delete an individual VA from the system.
View question list	View the list of questions used by your system to autodetect duplicates.

For further information relating to the identification of duplicates, please either ask your admin, consult *Configuration & Deployment* (page 6) for info on setting up or changing questions used for this process, or *Management Commands* (page 36) for admin guidance on running the process manually.

5.1.4 Repairing VA Errors & Warnings

Occasionally, you may need to edit an individual [VA](#), especially in cases where warnings or errors have been reported for the record. You can reach the Edit form from the Individual [VA](#) page by clicking the “Edit Record” button. Doing so transforms the [VA](#) responses into a series of form elements that are compliant to the [WHO VA](#) instrument standard and make data entry for specifically formatted values (such as datetime) easier. Questions that have already been answered will be pre-filled with their responses. You can expect to encounter the following types of form elements when editing the various responses to a [VA](#):

- **(Text Field)** For short responses
- **(Large Text Field)** For longer/ narrative-style responses
- **(Upload Selector)** For images and file attachments
- **(Datetime Selector)** For choosing a date and time response
- **(Radio Buttons)** For choosing 1 response from N (small #) choices
- **(Dropdown)** For choosing 1 response from N (large #) choices
- **(Calculated)** You are unable to edit these, they update according to other editable fields
- **(Checkboxes)** For selecting multiple responses to the same question

Using this form, you may edit the [VA](#) as needed. Upon finishing, scroll or navigate via the bottom right floating action button to the bottom of the page and click “Save” to confirm your edits.

After editing a [VA](#), the natural next step is to see if the changes have corrected any reported warnings or errors associated with the [VA](#). Data Managers can complete this second step in the repair process by clicking “Home” in the navigation bar to visit the Home page and starting a [VA](#) coding job to assign Cause of Death:

- Click the “Run Coding Algorithms” button to execute the coding algorithm in the background

If you receive any error messages following this, either reach out to your admin or see our [Troubleshooting](#) (page 42) section

5.2 Data Viewers

Data viewers have the simplest set of permissions, enabling them to view data scoped to whichever restrictions their admins set. As a data viewer, VA Explorer does not currently support any features unique to your role. Please consult the [Common Actions](#) (page 26) section, particularly the Dashboard and [VA](#) Search interface guides for more info on available actions.

5.3 Field Workers

Field Workers are similar to Data Viewers. However, they only see [VAs](#) for their own location and can partially repair [VA](#) Errors & Warnings for those [VAs](#). For a full listing of these, see [Common Actions](#) (page 26) or see below for an explanation of the “partial repair” workflow.

5.3.1 Partially Repairing VA Errors & Warnings

As a Field Worker, you have the ability to make corrections to [VAs](#) for your own location as a way of assisting Data Managers with the correction of any warnings or errors reported by VA Explorer. See [Repairing VA Errors & Warnings](#) (page 32) for details. However, you will not be able to create any [VA](#) coding jobs to re-run cause of death assignment. Please reach out to the appropriate Data Manager or Admin to do this for you.

THE ADMIN ROLE

6.1 User Admin

As admin, you can create and manage user accounts. Each of these accounts will have a role as described in *User Roles and Capabilities* (page 30). Additionally, there are workflows available to you as an admin to affect users as a group or as individuals described below.

6.1.1 Loading Locations for Geographic Access Support

To set up VA Explorer for the Geographic Access mentioned in that section, you must load location data into the system.

Locations in VA Explorer follow a hierarchical structure. A specific geographic region or jurisdiction has a name (“Name”), a type (“Type”), and a parent (“Parent”). By specifying the Parent field, you can achieve the arbitrary level of nesting required to make a tree.

The table below shows an example location hierarchy for States, Counties, and Cities in the United States. In this example, we have one state (California), two counties (Marin and Los Angeles), and three cities (Sausalito, San Rafael, and Los Angeles).

Table 1: An example geographic hierarchy in tabular format

Name	Type	Parent
California	State	
Marin County	County	California
Sausalito	City	Marin County
San Rafael	City	Marin County
Los Angeles County	County	California
Los Angeles	City	Los Angeles County

The input is similarly structured to support any number of geographic hierarchies for VA Explorer users. With a *CSV* file in hand, you can now supplement your initial system set up with the `load_locations` management command. Full usage details for this are provided in *Management Commands* (page 36).

Following this command, VA Explorer should support geographic restrictions to any area or facility you’ve provide, making them available during user creation and editing. Note that access to geographies higher up in the given tree equates to access for that geographic area as well as all its children-geographies. For example, in the above tree a user with access to California also has access to Marin County, Los Angeles County, Sausalito, San Rafael, and Los Angeles.



Fig. 1: A tree data structure showing the example geographic hierarchy from the previous table

6.1.2 Creating & Editing Users

Click “Users” in the navigation bar to visit the Users page. Click the “Create User” button to access the user creation form. Alternatively, click the “Edit” button in the table row for the user you would like to edit. Required fields are marked with an asterisk(*). To fill in the form:

1. Enter the user’s name
2. Enter the user’s email address. An email address can be used in the system once
3. Choose the user’s role from the dropdown menu
4. Check “Can View PII” if the user is allowed to view *PII* in the system. Also check “Can Download Data” if the should be allowed to export *VA* data from the system. See [Abilities to Download and View PII Data](#) (page 25) for details on these permissions if needed
5. Select the geographic region(s), facility, facilities this user can access. Users with a Field Worker role must be assigned to at least one facility

Click “Create” to create the user or “Update” if editing a user. “Cancel” will take you back to the Users page without completing the action.

When the “Create” button is clicked during the “Create” process or the “Update” button is clicked during the “Edit” process, VA Explorer validates the information you have entered. If there are errors associated with the provided data, VA Explorer will not create or update the user and the system will show error messages in red underneath the fields that require correction. Follow instructions to correct these errors and complete the user creation or edit process if needed.

6.1.3 Deactivating Users

User accounts are deactivated rather than deleted from VA Explorer. When a user account is deactivated, they will no longer be able to sign into the system. Their associated data, however, will still be visible. To deactivate a user, click “Users” in the navigation bar to visit the Users page. Click the “Edit” button in the table row for the user you would like to deactivate. Uncheck the “Active” checkbox. Click “Update.” The user will now be deactivated.

6.1.4 Bulk Creating Users

If you would like to create multiple users at once, particularly during initial system set up, VA Explorer provides that functionality as a management command. Please refer to [Management Commands](#) (page 36) for details on running `get_user_form_template` and `bulk_load_users` to obtain a bulk user creation template and to create your set of users based on that input file.

6.2 Data Admin

As admin, you have full access to all data-related actions. Each step of the process, from import to editing, to export has workflows available to you as an admin to affect it as needed.

6.2.1 Manually Importing Data

If you would like to manually import data from file, (perhaps if troubleshooting the automatic import process) VA Explorer supports this through the `load_va_csv` management command. Please refer to [Management Commands](#) (page 36) for details on usage.

Similarly, if you have configured VA Explorer to integrate with [ODK](#) and would like to manually import data from [ODK](#), VA Explorer also supports this. Use `import_from_odk` similarly detailed in [Management Commands](#) (page 36)

6.2.2 Manually Running Coding Algorithms

VA Explorer currently supports the InterVA5 coding algorithm and its associated settings to assign cause of death (*COD*) and may support other coding algorithms in the future such as InSilicoVA. The InterVA5 coding algorithm depends on the docker services as described in [Algorithm Support](#) (page 20)

To run InterVA5 manually, use `run_coding_algorithms` or see the entry for it in [Management Commands](#) (page 36) for full usage details. After the command has finished execution, a brief report of results should be printed to console used to run the management command (ex. Coded 24 verbal autopsies (out of 30) [6 issues]) or an error message if `pyCrossVA` or `InterVA5` are somehow unavailable; if that is the case, please refer to the [Troubleshooting](#) (page 42) section.

6.3 Management Commands

Beyond actions supported by the VA Explorer interface, there are a series of management commands available exclusively to developers, admins, and *IT* staff helping to maintain the service. To take advantage of these special commands, admins and *IT* staff need to, from the server hosting the VA Explorer instance, enter VA Explorer’s main Django container like so `docker exec -it va_explorer_django_1 bash`. From there `manage.py` is available for calling the following commands via:

```
manage.py <command> --<parameter_name>=<parameter_input>
```

from within the container. Below is a selection of management commands, generally useful to admins. An even fuller list of these can be found under [Development Commands](#) (page 49)

Table 2: Descriptions and lists of parameters for select management commands related to administration of VA Explorer

Command Name	Parameter Names	Description
	(*) = Required	
load_va_csv	--csv_file (*)	Used to manually import data from file to into VA Explorer's database. <code>csv_file</code> is a filename in the local folder or <code>unix:path</code> format location of the file. Can be used with <code>random_locations</code> for test or demo data to randomly assign the VA to a field worker with specific location restrictions. True or False; defaults to False
	--random_locations	
load_locations	--csv_file (*)	Used to load initial location data needed to support Geographic access. <code>csv_file</code> is a filename in the local folder or <code>unix:path</code> format location of the file. Can be used with <code>delete_previous</code> to delete existing location data and start fresh with the new locations being loaded. True or False; defaults to False
	--delete_previous	
run_coding_algorithms	--overwrite	Used to call supported algorithms for assignment of cause of death to all uncoded verbal autopsies. <code>overwrite</code> allows this command to clear (and save) all existing CoD assignments before running on every verbal autopsy regardless of whether it's coded or not. True or False; defaults to False <code>cod_fname</code> is a filename or <code>unix:path</code> format location to save the old CoDs to. Defaults to <code>old_cod_mapping.csv</code>
	--cod_fname	
get_user_form_template	--output_file	Utility to obtain a bulk user creation template csv with header fields corresponding to fields of the current User model. <code>output_file</code> is a filename or <code>unix:path</code> format location to save template to. Default is <code>user_form_fields.csv</code>
bulk_load_users	--user_list_file (*)	Used to bulk create user accounts, assigning a temporary password to each. <code>user_list_file</code> is a filename in the local folder or <code>unix:path</code> format location of the users file. Can be used with <code>email_confirmation</code> if an email server has been setup to automatically send an email with the new temporary password to each created user. True or False; defaults to False (prints to console instead so passwords must manually be passed to users somehow)
	--email_confirmation	

export_user_info	--output_file	Used to export an anonymized (No PII) list of all users in the system along with their roles and permissions. Useful for non-invasively correlating user activity in VA Explorer logs. <code>output_file</code> is a filename in the local folder or <code>unix:path</code> format location of the file to export user info to. Defaults to <code>user_list.csv</code> . <code>user_file</code> is a separate filename in the local folder or <code>unix:path</code> format location of a <code>.txt</code> file containing emails (one per line) of specific users to export. Defaults to all users if no file.
	--user_file	
mark_vas_as_duplicate	None	Used to manually run (or re-run if config is changed) duplicate checking within VA Explorer. Contains no parameters as behavior is determined by the configuration variable <code>QUESTIONS_TO_AUTODETECT_DUPLICATES</code> see Configuration & Deployment (page 6)

Additionally, if VA Explorer has been configured with integrations, the following additional management commands are available. If the environment variables (see [Integrations](#) (page 17)) that enable these integrations to work automatically are not defined, consider all parameters required for these management commands.

Table 3: Descriptions and lists of parameters for select management commands related to integrations

Command Name	Parameter Names	Description
import_from_odk	--email	Used to manually import VA data from ODK Central. Parameters are as described for the equivalent environment variables listed in Integrations (page 17) > ODK Central (page 18)
	--password	
	--project_name	
	--project_id	
	--form_id	
	--form_name	
import_from_kobo	--token	Used to manually import VA data from KoboToolbox. Parameters are as described for the equivalent environment variables listed in Integrations (page 17) > KoboToolbox (page 19)
	--asset_id	
load_dhis_cod_codes	--csv_file	Used to manually setup VA Explorer to report WHO CoD Codes in the format expected by DHIS2 and must be run before first export will succeed. <code>csv_file</code> is a filename in the local folder or <code>unix:path</code> format location of the csv file. csv file has the header <code>codsource,codcode,codname,codid</code>
run_dhis	--dhis_user	Used to manually export VA data to DHIS2. Parameters are as described for the equivalent environment variables listed in Integrations (page 17) > DHIS2 (page 19)
	--dhis_pass	
	--dhis_url	
	--dhis_orgunit	

IT SUPPORT

The VA Explorer team anticipates that there may be *IT* staff supporting the use of VA Explorer and recognizes that *IT* training for VA Explorer is different from that for day-to-day VA Explorer admins. The guides in this section outline estimated level of support for interested *IT* groups within organizations and to cover guidance this final type of user and their abilities. In some cases, the VA Explorer administrator may also serve as the *IT* staff.

7.1 Levels of Support Recommended for VA Explorer

This section will describe the levels of support within a country needed to install and maintain the VA Explorer system. While VA Explorer is pre-configured to run on deployment, VA Explorer integrates with other systems and having a system administrator who has the necessary skills to maintain the system over time is crucial to long term success. *IT* support can do more complex tasks than VA Explorer admins characterized by overall system installation, deployment, and maintenance that may occur outside of VA Explorer itself. The estimate of level-of-support described below is presented as reference only, and your jurisdiction may have specific needs that change these requirements.

Required:

- Basic understanding of/experience with Docker
- Copy files into/out of Docker containers and exec into/out of containers
- Change configuration settings/environment variables
- Experience with Linux systems and ssh

Nice To Haves:

- Basic knowledge of Django (particularly management commands)
- Familiarity with *ODK VA* form/ how to set up an *ODK* instance

Estimated time investment:

- Initial deployment: approximately 1 day to 1 week, depending on experience
- Maintenance: approximately 1-8 hours a week to debug and troubleshoot

Additionally, those *IT* teams wishing to contribute to the development of VA Explorer would also benefit from web development experience, particularly with Django. Development methods are discussed more in *Development* (page 47).

7.2 Backing Up VA Explorer

VA Explorer is distributed with utilities to backup data from the built-in database service. However, the server itself is something left to *IT* Staff/Admins to backup if desired. Some popular options for this are taking incremental backups of the server filesystem via snapshot utility Rsnapshot or even simple shell script.

Whichever method is chosen, establishing a regular backup method will help protect against critical loss of VA Explorer components such as your `.env` file, https certificates, reverse proxy configurations, and other items from previous installation and setup.

7.3 Upgrading VA Explorer

Those wanting to upgrade to the latest version of VA Explorer can do so easily via git, the same way they installed the software. Do another `git pull`, ensuring that any changes you or your organization have made such as configuring the `.env` file, are not erased.

If *IT* Admins need to migrate to a newer postgres database, as is occasionally the case, then that process is a bit more involved:

0. (optional) If existing migration volumes exist from a past upgrade, delete those now:

```
docker volume rm va_explorer_migration_postgres_data va_explorer_migration_postgres_data_
↳backups
```

1. Make a logical backup of all current data

```
docker exec -it va_explorer_vapostgres_1 /usr/bin/pg_dumpall -U postgres > ~/dumpfile
```

2. Make backups of old volumes by cloning data

- Get docker-compose labels

```
docker volume inspect va_explorer_production_postgres_data
```

```
docker volume create \
  --label com.docker.compose.project="va_explorer" \
  --label com.docker.compose.version="1.26.0" \
  --label com.docker.compose.volume="migration_postgres_data" \
  va_explorer_migration_postgres_data
```

- Clone data

```
docker container run --rm -it -v va_explorer_production_postgres_data:/from -v va_
↳explorer_migration_postgres_data:/to alpine ash -c "cd /from ; cp -av . /to"
```

- Repeat for backup volumes

```
docker volume inspect va_explorer_production_postgres_data_backups
```

```
docker volume create \
  --label com.docker.compose.project="va_explorer" \
  --label com.docker.compose.version="1.26.0" \
  --label com.docker.compose.volume="migration_postgres_data_backups" \
  va_explorer_migration_postgres_data_backups
```

```
docker container run --rm -it -v va_explorer_production_postgres_data_backups:/from -v
↪va_explorer_migration_postgres_data_backups:/to alpine ash -c "cd /from ; cp -av . /to"
```

3. Delete old volumes that no longer work with postgres version

```
docker volume rm va_explorer_production_postgres_data va_explorer_production_postgres_
↪data_backups
```

4. Build new VA Explorer release & rebuild fresh old volumes (no data)

```
docker-compose down && docker-compose build && docker-compose up -d
```

5. Copy over backup dumpfile data to new container

```
docker cp ~/dumpfile va_explorer_vapostgres_1:/tmp/dumpfile
```

```
docker exec -it va_explorer_vapostgres_1 bash
```

```
psql -U postgres
```

Note: DROP DATABASE may fail as the new container does migrations, just re-try

```
DROP DATABASE va_explorer;
CREATE DATABASE va_explorer;
```

```
psql -U postgres < /tmp/dumpfile
exit
```

6. Run docker-compose restart
7. Confirm that all data made it / backup & restore appears successful
8. Use the old volumes va_explorer_migration_postgres_data and va_explorer_migration_postgres_data_backups to rollback if needed.

See also any release notes associated with new VA Explorer versions that may contain version-specific upgrade instructions in the future.

TROUBLESHOOTING

Not everything can go right all the time. If VA Explorer is acting in an unexpected way, troubleshooting can help determine, and perhaps solve, the problem.

8.1 Reading and Interpreting VA Explorer Logs

Logs are often the most helpful to operators, and even developers fielding questions, when attempting to diagnose a potential issue.

- For local instances of VA Explorer, logs should be readily available and printing to the console of the program used to run `runserver_plus`
- For operators, you can view logs through Docker. Run `docker logs --tail <quantity> va_explorer_django_1` where quantity can be something like 100 to see the last 100 lines of logs or 1m to see the last minute.

As mentioned in *Configuration & Deployment* (page 6), `DJANGO_DEBUG` can be set to `True` for even more useful troubleshooting information in those logs. Running commands like `docker ps` will also help you determine if all docker services are running as expected.

8.2 Frequently Asked Questions

1. Is VA Explorer free?

Yes, VA Explorer is free to download and use as you please under the terms of the open-source Apache V2 license. If installed on a linux-based *OS*, users also do not need to worry about any costs associated with using docker. We expect the only costs associated with VA Explorer to be hosting and domain name registration.

2. I'm getting an error related to `psycpg2` when setting up VA Explorer. What does this mean?

It is possible that `psycpg2` is pointing at the wrong SSL installation on your system it is attempting to set itself up during the pip install process. Adding this environment variable has worked as a potential fix:

```
export LDFlags='-L/usr/local/lib -L/usr/local/opt/openssl/lib -L/usr/local/opt/readline/  
↳lib'
```

3. Why is the amount of VAs in VA Explorer is different from the amount of data from wherever I imported them?

This may appear in a variety of contexts, so the solution may differ:

- If referring to the **import process**, the numbers rely on each VA having a unique `instanceid` (typically in uuid format) and a unique `instancename` (typically constructed from the name of the deceased and date of the interview). The import process does its best to recognize changes to these during imports and react accordingly, but if both change at once (for example, when editing the name of the deceased causes the `instanceid` to update and a new `instancename` to be constructed) then duplicate data may result with the old name of the deceased. It is recommended to take advantage of provided [Data Cleanup Operations](#) (page 31) to identify and delete these.
- If referring to the **dashboard**, the numbers likely differ because the dashboard only shows VAs that have been assigned a cause of death. To make the numbers match, either sum the “Coded VAs” and “Uncoded VAs” values, or run the coding algorithms again to assign causes of death to any VAs that are missing it. Please note, VAs with coding issues (highlighted on the homepage) will not code correctly until the error is corrected.

4. I’m getting an error related to scipy when setting up VA Explorer. What does this mean?

Some MacOS users have reported getting the error `numpy.distutils.system_info.NotFoundError: No lapack/blas resources found`. Note: Accelerate is no longer supported. If you get this error, this [thread](#)¹⁶ may be a helpful resource. Ensuring that pip is upgraded `pip install --upgrade pip` may also help.

5. What is an easy configuration to use for my reverse proxy?

This is a simple example apache config that utilizes letsencrypt https certificates

```
<VirtualHost *:80>
    ServerName myhost.com
    ServerAdmin admin@myhost.com
    ServerAlias *.myhost.com

    Redirect permanent / https://myhost.com/

    ErrorLog ${APACHE_LOG_DIR}/http-myhost-error.log
    CustomLog ${APACHE_LOG_DIR}/http-myhost-access.log combined
</VirtualHost>

<IfModule mod_ssl.c>
    <VirtualHost *:443>
        ServerName myhost.com
        ServerAdmin admin@myhost.com
        ServerAlias *.myhost.com
        SSLProxyEngine on

        ProxyPreserveHost on
        ProxyVia on
        ProxyPass / http://localhost:5000/
        ProxyPassReverse / http://localhost:5000/

        ErrorLog ${APACHE_LOG_DIR}/https-myhost-error.log
        CustomLog ${APACHE_LOG_DIR}/https-myhost-access.log combined

        SSLCertificateFile /etc/letsencrypt/live/myhost.com/fullchain.pem
        SSLCertificateKeyFile /etc/letsencrypt/live/myhost.com/privkey.pem
        Include /etc/letsencrypt/options-ssl-apache.conf
    </VirtualHost>
</IfModule>
```

6. Why are my users not receiving email?

¹⁶ <https://github.com/scipy/scipy/issues/13102#issuecomment-962468269>

Email technology can unfortunately encounter issues delivering messages sometimes. By default, VA Explorer prints to the console and can be configured to work with an existing mail server. This mail server, once VA Explorer is configured to use it may not work if:

- You or your organization are using a cloud provider that restricts the usage of simple mail servers as a spam-prevention strategy. The cloud provider may offer a mail service you can use instead
- You or your organization could be assigned an IP address that was previously used for sending spam and is therefore blocked by many mail recipients
- You or your organization's domain may not be recognized by mail recipients, so messages from it may be blocked or marked as spam

If you want to troubleshoot sending emails from your mail server, the mail-tester service recommended by [ODK Central](#) may help identify your email delivery issues.

Part III

Development

DEVELOPMENT

Thank you for your interest in helping to make VA Explorer even better! These guides should help you get setup for local development. Afterwards, you should be ready to tackle [Issues](#)¹⁷. We recommend the ones tagged `good first issue` as a starting point.

As a pre-requisite you should have the following already on your system:

- Python 3 + pip
- Postgres
- Docker

9.1 Setting Up & Building

Like the process described in the VA Explorer `README.md`:

1. Retrieve the application source code

```
git clone https://github.com/VA-Explorer/va_explorer.git
```

2. Change into the new directory

```
cd va_explorer
```pref
```

3. Create a virtual env

```
```shell
python -m venv venv
```

4. Activate the virtual env:

```
source venv/bin/activate
```

5. Install application requirements

```
pip install -r requirements/base.txt
```

6. Create the `va_explorer` database using your postgres user made during postgres download. It may be postgres for example.

¹⁷ https://github.com/VA-Explorer/va_explorer/issues

```
createdb va_explorer -U <name of Postgres user> --password`
```

7. Create a .env file at the project root with the following key/value pairs:

```
DATABASE_URL=psql://<YOUR POSTGRESUSER>:<POSTGRESUSER PASSWORD>@localhost/va_explorer  
CELERY_BROKER_URL=redis://localhost:6379/0
```

8. Run the database migrations

```
./manage.py makemigrations  
./manage.py migrate
```

This will prepare your local development environment to run VA Explorer locally via the `runserver_plus` command (see [Development Commands](#) (page 49)). Next you will want to seed VA Explorer with some example data so you can login and see its features in action. To do that run:

1. Create user roles & permissions source code

```
./manage.py initialize_groups
```

2. Create an admin user for yourself. The values can be fake, you just need to remember them to login

```
./manage.py seed_admin_user <EMAIL_ADDRESS> --password=<PASSWORD>
```

3. Create some demo accounts if you'd like to try out the other roles too

```
./manage.py seed_demo_users
```

4. If you have locations for geographic access restrictions on hand, load those via

```
./manage.py load_locations <NAME OF CSV>
```

5. If you have [VAs](#) on hand, you should also load them now

```
./manage.py load_va_csv <NAME OF CSV>
```

6. Finally, if you'd like to try out the coding algorithm assignment functionality, build just those docker services and run them manually via:

```
docker-compose up -d --build pycrossva interval5  
./manage.py run_coding_algorithms
```

You can also run the rest of the docker build, the part of VA Explorer that is deployed for users, by running `docker-compose up -d --build` to build the rest of the containers.

9.2 Testing & Running Locally

After setup, you're ready to run VA Explorer locally! If you've run:

```
./manage.py runserver_plus 0.0.0.0:8000
```

then you should be able to navigate to `localhost:8000` in the browser of your choice and be presented with the sign in screen. Signing in here with the admin user you seeded earlier should open up access to the rest of the features, as described in [Features](#) (page 11).

Note: Data seeded for local instances of VA Explorer lives in your local installation of PostgreSQL, so you should be able to examine the data by running `psql -U postgres` and connecting to the `va_explorer` database.

If you're trying out the built docker version of VA Explorer locally, you should be able to navigate to the docker deployment of VA Explorer by going to `localhost:5000` in your browser after building. Unlike your local instance, data seeded for this will only exist if you `docker exec` into the `django` docker service. That data lives in the `postgres` docker service, separate from your local installation of PostgreSQL.

Finally, developers should be able to test easily and often with the built-in testing utility. To run the test suite use `pytest`. When running these, you may be required to install Firefox or `geckodriver` to do browser-based tests. See the Selenium Client Driver Documentation for more info if needed.

9.3 Running Integrations Locally

If you would like to test or contribute to the functionality of VA Explorer integrations and would like to use the real instances of *ODK Central* and *DHIS2* locally, please see to the documentation from those two services. They are the best reference.

- *ODK Central*¹⁸
- *DHIS2*¹⁹
 - via Docker²⁰

9.4 Development Commands

As mentioned in *Management Commands* (page 36), VA Explorer provides additional functionality beyond its UI-based features. This holds especially true for development which has relevant actions detailed here:

¹⁸ <https://docs.getodk.org/central-intro/>

¹⁹ <https://developers.dhis2.org/docs>

²⁰ <https://hub.docker.com/r/dhis2/core>

Table 1: List of all configuration options when interfacing with the py-CrossVA service, plus default values and descriptions for each

Command Name	Parameter Names	Description
	(*) = Required	
<code>makemigrations</code>	See Ext. Docs	A built-in django admin command used to generate new migration files based on changes detected to VA Explorer models.
<code>migrate</code>	See Ext. Docs	A built-in django admin command used to sync the VA Explorer database state with the current set of models and migrations in the app code by applying migrations files.
<code>shell_plus</code>	None	A django-extensions command that provides a python shell session preloaded with VA Explorer database models and other user-defined classes. Useful for running python commands that require VA Explorer components.
<code>collectstatic</code>	See Ext. Docs	A built-in django admin command that collects all static files into <code>STATIC_ROOT</code> . Useful for refreshing the application during development after adding new static files.
<code>runserver_plus</code>	<code>ip:port</code> or See Ext. Docs	A django-extensions command that runs a development server to serve VA Explorer locally. Has extras that support errors and debugging. Optionally specify an <code>ip:port</code> format location to run the server on a specific IP address and port number. Defaults to <code>127.0.0.1:8000</code>
<code>validate_templates</code>	See Ext. Docs	A django-extensions command used to catch any invalid Django template syntax within your app. Used by VA Explorer in its testing.
<code>initialize_groups</code>	None	Used to manually run or update user roles and their associated permissions.
<code>seed_admin_user</code>	<code>--email (*)</code>	Creates an admin user with a specific email address and password. Optionally accepts a password via <i>password</i> param, but will create a randomly-generated password and print to the console by default.
	<code>--password</code>	
<code>seed_demo_users</code>	None	Creates demo accounts for Data manager, Data viewer, and Field Worker with Password1. Used for the local environment only for demonstration and testing purposes.
<code>fake_current_va_dates</code>	None	Updates dates on all VAs to make them look more current. Used for the local environment only for demonstration and testing purposes.
<code>randomize_va_dates</code>	None	Initializes demo VAs with dates. Used for the local environment only for demonstration and testing purposes.

Additionally, the full and complete list of management commands (only some of the most popular are described here or in [Management Commands](#) (page 36)) is available by running `manage.py help`.

10.1 Security & Privacy

VA Explorer uses open-source third-party libraries from well-known and/or reputable sources. The MITRE Corporation does not vet the security of those software libraries. If security concerns you, you or your organization may want to review the third-party libraries and their source code on GitHub.

10.2 Attribution

This project relies on contributions from the community, in addition to other existing VA open-source libraries, without which VA Explorer would not be possible. Portions of this code, therefore, are copyright the respective developers, which include, but may not be limited to, teams from the CDC Foundation (e.g., DHIS-2 export) and The Ohio State University (e.g., InterVA5, pyCrossVA, openVA pipeline).

10.3 Copyright & License

Copyright © 2020-2023, The MITRE Corporation. Licensed under Apache 2.0

The source of this information is the Data for Health Initiative, a joint project of the CDC Foundation and Bloomberg Philanthropies.

Available publicly via [Apache License, Version 2.0](#)²¹. VA Explorer also includes external software that is available under a variety of licenses. See [LICENSE](#)²² for full license text(s).

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

²¹ <https://opensource.org/licenses/Apache-2.0>

²² https://github.com/VA-Explorer/va_explorer/blob/main/LICENSE

10.4 Contact

Contact verbal-autopsy@mitre.org with any questions or comments.

10.5 Other VA Resources

- [VA Standards Group at WHO](#)²³
- Other VA IT Tools
 - [OpenVA.net](#)²⁴
 - * [VA Software on GitHub](#)²⁵
 - * [OpenVA Pipeline](#)²⁶

Is your resource missing? Please let us know, and we'll include it!

10.6 Glossary

CCVA

Computer-Coded Verbal Autopsy

COD

Cause of Death

CRVS

Civil Registration and Vital Statistics

CSV

Comma Separated Value

DB

Database

DHIS2

District Health Information System, Version 2

IT

Information Technology

JSON

JavaScript Object Notation

ODK

Open Data Kit

OS

Operating System

PII

Personally Identifiable Information

²³ <https://www.who.int/standards/classifications/other-classifications/verbal-autopsy-standards-ascertaining-and-attributing-causes-of-death>

²⁴ <https://openva.net/#info>

²⁵ <https://github.com/verbal-autopsy-software>

²⁶ https://github.com/verbal-autopsy-software/openva_pipeline

VA

Verbal Autopsy

WHO

World Health Information

Last Updated: 2023-10-02

INDEX

C

CCVA, [52](#)

COD, [52](#)

CRVS, [52](#)

CSV, [52](#)

D

DB, [52](#)

DHIS2, [52](#)

I

IT, [52](#)

J

JSON, [52](#)

O

ODK, [52](#)

OS, [52](#)

P

PII, [52](#)

V

VA, [53](#)

W

WHO, [53](#)