TEK DB Technical Information and Maintenance Manual

Dependency Stack

- Operating System (pick 1):
  - **Ubuntu 22.04 LTS** (recommended/supported, 20.04 and newer LTS)
  - Other Linux Server distributions
  - Windows Server
- Web Server (pick 1):
  - **NGINX** (recommended/supported)
  - Apache
  - IIS (for Windows, w/FastCGI)
- Database:
  - **PostgreSQL with PostGIS extension**
- Language:
  - **Python 3.8+**
- Framework:
  - **Django 3.2**

Code Source

All code and documentation is available for free (under the MIT license) on Github:https://github.com/Ecotrust/TEKDB

Exporting the data

Automated Export for Backup

In 2022, a new feature was added to simplify the process below – it allows users with administrative privileges to export a zipped archive file containing a 'fixture' file to restore database records, as well as a folder containing all relevant media files (videos, photos, audio, pdfs, etc…) supported by the database.

First, the user must log in using their admin credentials. Next they should click on the menu button at the top right, which should have their username on it. From that button, they should find an option in the dropdown menu saying 'administration' which will take them to the Django Administration dashboard.
In the top right corner they should have a button that reads 'Export to .zip', alongside an 'Info' button, which will remind them about the sensitivity of the file they are about to download: all of the traditional knowledge that has been accumulated into the database thus far, captured in a single, unencrypted file.

Manual Export for Backup

If you are reinstalling the tool and need to preserve or migrate your data (not the same as migrating from the old Microsoft Access-based MTKEDB) you can create a 'fixture' (a backup file) that can be imported into new instances of the TEK DB. To do so:

- Decide on a safe place to write the file to
- Decide on a meaningful name for the file, like tek_db_fixture_YYYYMMDD.json
    - where YYYYMMDD is today's date
    - json is the data format. If you don't name it with the .json extension this process will still work.
- Navigate to your project and activate your virtual environment:
    - Linux

```
cd /usr/local/apps/TEKDB
source env/bin/activate
cd ./TEKDB
```

    - Windows

```
cd C:\Apps\TEKDB
env\Scripts\activate.bat
cd .\TEKDB
```

- Then run this WHOLE command to dump the data (the order is very specific to protect data dependencies) replacing the file at the end after the > with the file you chose above

```
python manage.py dumpdata --indent=2 auth.Group Lookup.LookupPlanningUnit
Lookup.LookupTribe Lookup.LookupHabitat TEKDB.Places
Lookup.LookupResourceGroup TEKDB.Resources Lookup.LookupPartUsed
```

```
Lookup.LookupCustomaryUse Lookup.LookupSeason Lookup.LookupTiming
Relationships.PlacesResourceEvents Lookup.LookupParticipants
Lookup.LookupTechniques Lookup.LookupActivity TEKDB.ResourcesActivityEvents
Lookup.People Lookup.LookupReferenceType Lookup.LookupAuthorType
TEKDB.Citations Relationships.PlacesCitationEvents Lookup.CurrentVersion
Lookup.LookupLocalityType TEKDB.Locality Relationships.LocalityGISSelections
Relationships.LocalityPlaceResourceEvent Lookup.LookupMediaType
Lookup.LookupUserInfo TEKDB.Media Relationships.MediaCitationEvents
Relationships.PlaceAltIndigenousName Relationships.PlaceGISSelections
Relationships.PlacesMediaEvents Relationships.PlacesResourceCitationEvents
Relationships.PlacesResourceMediaEvents
Relationships.ResourceActivityCitationEvents
Relationships.ResourceActivityMediaEvents
Relationships.ResourceAltIndigenousName Relationships.ResourceResourceEvents
Relationships.ResourcesCitationEvents Relationships.ResourcesMediaEvents
Accounts.UserAccess Accounts.Users explore.PageContent >
/path/to/your/file.json
```

Updating The Code

The code is all on GitHub and available to the public. Git, a tool for sharing the code and tracking changes made to it is already installed on your server. With these two pieces in place, getting code updates fairly easy, but can contain risks:

- It is **highly** recommended that you take a snapshot of your server before doing this.
- If you are not sure what new features you want out of this upgrade, there is no need to perform an update. If you are unsure, collaborating with Ecotrust we could sort out what features have been added since your last update: ksdev@ecotrust.org .
- Major risks:
    - New code will include database migrations that did not anticipate types of data in your instance. This could result in loss of data.
    - Local changes could have been made to your code base, resulting in a 'merge conflict'. Without familiarity with 'Git' these can be difficult to resolve, and guessing can quickly break things beyond repair.
    - Restoring from backup will fix both of these issues, but will prevent you from installing new updates. If you hit any of the issues above, please

contact Ecotrust and ask for the Software Developers at ksdev@ecotrust.org.

Automatic Update Script

As of 2022, there are new scripts (one for Linux installations, one for Windows) that will perform the following tasks for you, assuming:

- You created your Python virtual environment with the name 'env' and it lives in the top level of your git-cloned repository.
- You have git installed for the user running the script
- The user running the script has permissions to read/write/execute
    - Your project folder
    - Your application servers (IIS or NGINX)
    - Git
- If on Windows, the executable to restart IIS is at `C:\Windows\System32\iisreset.exe`

If all is installed correctly, you should be able to just run one of the following scripts:

- For Linux:
    - /path/to/app/TEKDB/scripts/Linux/update.sh
- For Windows:
    - \path\to\app\TEKDB\scripts\Windows\update.bat

The script performs the following tasks:

1. Pulls the latest code updates for your current branch of the Git repository
2. Installs any required updated python packages
3. Applies any updates to your database schema (new or altered tables, columns, etc…)
4. Prepares static files to be served by the application server
5. Restarts your servers:
    a. Linux – Uwsgi + NGINX
    b. Windows – IIS

Manual Update Procedure

1. Log in to your server
2. Change directories to TEK's code:
    i. Linux: `cd /usr/local/apps/TEKDB`
    ii. Windows: `cd C:\apps\TEKDB`
3. Retrieve all code updates from GitHub:
    i. `git fetch`
    ii. `git merge origin/main`
4. Activate your virtual environment:
    i. Linux: `source env/bin/activate`
    ii. Windows: `env\Scripts\activate.bat`

5. If 'manage.py' is not in the current folder, cd into the TEKDB folder that does have it (most likely ./TEKDB)
6. Run `pip install -r requirements.txt`
    i. If running Linux, also run `pip install -r requirements_linux.txt`
7. Run database migrations: `python manage.py migrate`
    i. If new tables are added or old tables are changed this should update your database schema and transform any necessary data appropriately, but this comes with some risk to your data, hence the backup recommendation.
8. Collect 'static files' to be served by the application: `python manage.py collectstatic`
9. Compress eligible 'static files' for quicker service: `python manage.py compress`
10. Restart servers:
    i. Linux:
        i. Restart uWSGI: `sudo service uwsgi restart`
        ii. Restart NGINX: `sudo service nginx restart`
    ii. Windows:
        i. Restart IIS: `C:\Windows\System32\iisreset.exe`

OS Security Updates [Ubuntu Linux]

Ubuntu Linux can be set up to download and install security patches by default, and a cron job can be created to restart them during convenient hours if needed.

If your server is not to be made accessible via the internet, and you trust all users logging on to your network, you may prefer not to download OS updates, as they can also (very rarely) be a source for corruption.

If you would like to do this by hand on Ubuntu Linux, do the following:

Query for all available updates:

```
sudo apt update
```

List all available updates:

```
sudo apt list --upgradable
```

Install all available updates:

```
sudo apt update
```

Install a single package with it's available update:

```
sudo apt install {PACKAGE_NAME}
```

If you are managing a Windows server, it is assumed that you chose this out of proficiency with Windows, and that you have the knowledge to update your server yourself.

Restart the server if prompted and server traffic is anticipated to be light.

Services

A deeper dive into some of the most important and dynamic pieces of the tool stack. Please note that commands below assume you are on Ubuntu Linux, and you may need to find another command to perform the same task on other operating systems. Links to each tool's website are provided in case more information is needed.

PostgreSQL & PostGIS

The database server, if causing problems, can be restarted with `sudo service postgres restart`.

On Linux, you can find most of PostgreSQL's files here: /etc/postresql

By default, logs will be kept here: /var/log/postgresql/

For more information, check out the PostgreSQL website and the PostGIS website.

It is expected that if the organization installing and running this database tool has a GIS department, they will want to connect to the data in the tool directly. There are some limits to this access, which should also be covered in the GIS User Guide.

1. If you install on Windows, connecting to the PostgreSQL/PostGIS database from ArcGIS may be difficult, or even disabled.
2. If the database is installed on Linux, ArcGIS users will be able to access the data as 'read only' – they can view it, they can save and edit it locally for their purposes, but they cannot add to or edit the data that is inside the database directly.
3. If the GIS team uses other GIS tools, such as QGIS, they very well may be able to add and edit data in the database directly.

NGINX

The default recommended HTTP server for Linux installs. Can be restarted with `sudo service nginx restart`

NGINX's files can be found in /etc/nginx

Configuration is managed in /etc/nginx/sites-available/tekdb

- A symlink to this file exists in /etc/nginx/sites-enabled/ if the site is live

Both access and error logs are held in /var/logs/nginx/ by default

For more information, check out the NGINX website.

uWSGI

A service enabling Django projects (and other web app frameworks) to be served by HTTP servers.

Can be restarted in Linux with `sudo service uwsgi restart`

For more information, check out the uWSGI Project website.

IIS

The default recommended server for Windows installations of this web application.

Other Server Management Commands [Ubuntu Linux]

Shutdown:
`sudo shutdown 0`

Restart:
`sudo reboot`

Show Running Processes and Hardware Usage:
`htop`
Note: this is an installed app that does not come with the operating system. It is an easier-to-read version of `top` which should come with the OS by default.

Update the ITK Database application:
`/path/to/app/TEKDB/scripts/Linux/update.sh`

This is most likely: /usr/local/apps/TEKDB/scripts/Linux/update.sh

Troubleshooting

Lost Django Admin Password

Only Admin users can create or edit users, and this includes passwords. If no administrator is able to login, this can be alarming, but it is easily fixed from the back end.

- Log in to the server
- Navigate to the project folder (likely /usr/local/apps/TEKDB)
- Activate the virtual environment `source env/bin/activate`
- Navigate to the manage.py file (either here or in ./TEKDB)
- Create a new Super User with `python manage.py createsuperuser`
    - You will be prompted to provide a username and a password
- Log in to the web application with the new Super User and reset the Admin passwords to something they will remember

Site is Down

- Is the error:
    - **404 - not found**:
        - Check that NGINX is running with `sudo service nginx status`
            - replace `status` with `start` if not
        - Check the NGINX logs for clues
        - Check the PostgreSQL logs
        - Check the (virtual) hardware:
            - Is the hard drive full (can happen from old logs building up or lots of media served from the tool)
                - See if you can't clear enough space by deleting old log files
                - Increase the hard drive capacity
        - Check the network:
            - are you on the same network as the tool?
            - does the given URL point to the server's IP address in your DNS records?

        - is NGINX configured to listen to requests to the given url?
        - is NGINX listening on port 80 (or whichever port the request is being made to)
    - If all else fails, turn it off and on again.
- **A page loads, but tool doesn't show**:
    - Check network configuration (as above)
    - Check NGINX is running
    - Check NGINX logs to confirm it is receiving the requests
    - Check NGINX config to see that it is correct
    - Check uWSGI is running and configured correctly
    - Restart services one at a time and test in this order:
        - uWSGI
        - NGINX
        - The server itself
- **A Django Error** (yellow header with gray debug information):
    - NOTE: this page should not show - be sure to set DEBUG = False in your settings.py and local_settings.py files
    - Review the page content for hints.
    - Review the NGINX docs for hints.
    - Restart uWSGI
    - Reboot the server
    - Create an issue on the GitHub project describing the error and how it was triggered.
- **A Plain Error Page** (Bold black header, plan black text on white background):
    - Follow same steps as 404 - Not found error
- **Page content appears but is mangled**:
    - Does page still look mangled on a wider browser window?
    - No images or improper page layout:
        - from the virtual environment and project folder, run `python manage.py collectstatic`
        - `Ensure that the file TEKDB/TEKDB/settings.py contains `DEBUG=False``
        - `Ensure that the file TEKDB/TEKDB/local_settings.py DOES NOT contain `DEBUG=True``
    - Refresh browser cache