

---

# **Terminator Documentation**

***Release 0.2.1***

**Leandro Regueiro Iglesias**

**Aug 30, 2017**



---

## Contents

---

<b>1</b>	<b>Install Terminator</b>	<b>3</b>
1.1	Local installation . . . . .	3
1.2	Deploying Terminator using a Web Server . . . . .	4
<b>2</b>	<b>User's guide</b>	<b>5</b>
2.1	User Roles . . . . .	5
2.2	How Terminator Handles Terminology . . . . .	5
2.3	Export . . . . .	6
2.4	Import . . . . .	6
2.5	Search for Terminology . . . . .	6
<b>3</b>	<b>Features</b>	<b>7</b>
3.1	Supported file formats . . . . .	7



Terminator is a web app that allows you to manage terminology more easily. Terminator is GPLv3 licensed Free Software, and you can download it and run your own copy if you want to.

You can also help or participate in the Terminator development in many ways, for example:

- reporting bugs,
- requesting features,
- translating it,
- contributing code,
- contributing documentation,
- testing it,
- ...

The Terminator project itself is hosted at <https://github.com/translate/terminator/> where you can find details about source code, mailing lists, etc.

Contents:



# CHAPTER 1

---

## Install Terminator

---

Want to try Terminator? This guide will guide you through installing Terminator and its requirements.

Before proceeding, consider installing these first:

- At least Python 2.6
- `python-pip` installer
- `git` distributed version control system
- A database management system like PostgreSQL or MySQL
- The Python bindings for the installed database management system
- Optionally a mail server

## Local installation

In order to install Terminator run the following commands:

```
$ sudo pip install virtualenvwrapper
$ mkvirtualenv <env-name>
(env-name) $ git clone https://github.com/translate/terminator.git
(env-name) $ cd terminator
(env-name) $ pip install -r requirements/base.txt
```

## Creating a Database

It is necessary to create a database to store all Terminator data. This **database must be UTF-8 encoded**. You can give it any name. How to create that database is out of this documents scope, so please refer to your database documentation.

**Warning:** This database must be UTF-8 encoded.

## Customizing the Configuration

The initial configuration includes settings that most likely you are going to change, for example you must change the *DATABASES* setting to match your database settings (database engine, database name, database user,...), or change the *DEBUG*, *SECRET\_KEY*, *TIME\_ZONE*, *EMAIL\_PORT* or *ADMINS* settings.

---

**Note:** If you are going to use a mail server perhaps you will have to customize other Django settings not included in Terminator configuration. If this is the case then just add those settings to Django configuration file.

---

## Setting Up the Database

Before your run Terminator for the first time, you need to create the schema for the database and populate it with initial data. This is done by executing the *syncdb* management command:

```
(env-name) $ cd project
(env-name) $ python manage.py syncdb --noinput
```

## Running Terminator

To run it, just issue:

```
(env-name) $ python manage.py runserver
```

And the server will start listening on port 8000. This can be accessed from your web browser at `http://localhost:8000/`.

This default installation already provides several default user accounts with different roles:

- User **usuario** with password *usuario* (has superuser role)
- User **dono** with password *dono*
- User **lexi** with password *lexi*
- User **term** with password *term*

If you want to try the import feature you have to use the *usuario* user account. You may test it using TBX files like the ones provided by [Proxecto Trasno](#).

## Deploying Terminator using a Web Server

If you want to deploy Terminator using a web server like Apache, please refer to [Django deployment documentation](#).



Terminator is a web app that allows you to manage terminology more easily. Despite the simplicity of its purpose, Terminator can be considered a somewhat complex tool, so this user guide is intended to help people learning how to correctly use it.

## User Roles

Terminator has six different user roles with increasing permissions (each role inherits previous role permissions):

- *Anonymous user*: those users that are not authenticated.
- *Authenticated user*: users authenticated but without any permission on any glossary.
- Terminologist:
- Lexicographer:
- Glossary owner:
- Server administrator:

## How Terminator Handles Terminology

Terminator encloses all terminology in glossaries. Terminology is managed in a concept-centric way, allowing several terms for each concept, even for the same language.

## Export

### Export to TBX

Terminator is capable of exporting to a subset of the ISO 30042 TermBase eXchange (TBX) standard. See [tbx-conformance](#) for more details.

## Import

### Import from TBX

Terminator is capable of importing from a subset of the ISO 30042 TermBase eXchange (TBX) standard. See [tbx-conformance](#) for more details.

**Warning:** Terminator assumes that some elements that are used in the TBX file, for example the languages, are already created in Terminator. If those elements don't exist in Terminator then the import will fail showing an error message.

**Warning:** Terminator assumes that some required elements are present in the TBX file, thus if those elements are not present then the import process will fail showing an error message. See [tbx-conformance](#) for more details on the required elements.

---

**Note:** If any other element, besides the supported by Terminator ones, is present in the TBX file Terminator won't import those, but the importing process won't fail. See [tbx-conformance](#) for more details on which are the supported elements.

---

## Search for Terminology

Terminator has plenty of features.

### **Supported file formats**

Terminator supports ISO 30042 TermBase eXchange (TBX) format, but only part of its features. See [tbx-conformance](#) for more details.