



# **Azur Documentation**

*Release 1.0-dev*

**Harold Erbin**

May 14, 2011



# CONTENTS

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Installation</b>	<b>3</b>
2.1	Get Azur . . . . .	3
2.2	Configuration . . . . .	3
2.3	Tests . . . . .	3
<b>3</b>	<b>Tools details</b>	<b>5</b>
3.1	Summary . . . . .	5
3.2	Fantasy/SF . . . . .	6
3.3	Linguistic . . . . .	7
3.4	Projects . . . . .	7
3.5	Writing . . . . .	7
<b>4</b>	<b>User documentation</b>	<b>9</b>
4.1	Markup . . . . .	9
<b>5</b>	<b>Code documentation</b>	<b>11</b>
5.1	core — Core modules . . . . .	11
5.2	settings — Django settings . . . . .	12
<b>6</b>	<b>Glossary</b>	<b>13</b>
<b>7</b>	<b>Authors</b>	<b>15</b>
<b>8</b>	<b>Thanks</b>	<b>17</b>
<b>9</b>	<b>Copyright</b>	<b>19</b>
9.1	Licence . . . . .	19
<b>10</b>	<b>Changes in Azur</b>	<b>21</b>
10.1	Release 1.0 (in development) . . . . .	21
<b>11</b>	<b>Indices and tables</b>	<b>23</b>
	<b>Python Module Index</b>	<b>25</b>
	<b>Index</b>	<b>27</b>



# INTRODUCTION

Azur is a set of tools to help writers — and also amateurs of roleplay — to construct their universe and manage their projects. Its aim is to be like a *forge*, but for writers.

The aim of this software is to help the writer(s) to construct a complete universe by checking his coherence and simplify his work: nowadays there is exists no software which aggregate all the tools a writer could need (even if there are software adapted for some parts, like wiki, but it's not convenient).

The main features are (see *Changes in Azur* to know what is already implemented):

- cards for characters (name, biography. . .);
- relationships between characters (family, enemies, schools. . .);
- events and timeline;
- description of the universe (objects, cultures. . .), using wiki-like pages;
- planetary systems;
- geodesy, geography and time conversions;
- linguistic (words generators using grammar rules. . .);
- “bugtracker”;
- books management (list, statistics. . .);
- public informations;
- collaborative work (chat. . .).

Some modules work together and it's possible to extend by writing new modules (e.g. adding races for characters in fantasy books).



# INSTALLATION

## 2.1 Get Azur

You can find the source code in the Mercurial repository, hosted on [Bitbucket](#).

If you discover a bug or you have some ideas to improve Azur, you can submit a [new report](#) in the bugtracker on Bitbucket.

## 2.2 Configuration

In order to install Azur, you must fill the file `settings.py` to fit your Django installation.

## 2.3 Tests

It is easier to use a SQLite database for testing purposes.





# TOOLS DETAILS

Check the summary file first. The other sections contains more informations on some topics.

## 3.1 Summary

### 3.1.1 Physics

These tools are mainly intended for SF and fantasy projects.

- Simulation of all elements of a stellar systems (star, planets, satellites, comets...) with calculation to deduct other parameters or information (habitable zone <sup>1</sup>, mountain maximal height...).
- Geodesic (e.g. computation of distance and itinerary on an object).
- Atmospheres computations (temperature on surface, sky colours...).
- Description of the celestial sphere.
- Helpers to manage our own unit systems.
- Plot the 2D and 3D map of the objects with the data.

### 3.1.2 Dates, calendars and events

- Creation of calendars.
- Advanced date formatting and utilisation, based on the context.
- Date conversions between calendars (from different planets or not).
- Management of events, with bindings to characters and relationships.
- Compute time to travel between places of different events for a character.

### 3.1.3 Characters

- Creation of characters (moral and physical portraits, history...).
- Various relationship management between characters (family, friends, groups, schools...).
- Verifications of informations (date of birth with parent longevity...).

---

<sup>1</sup> System region where liquid water can be found.

### 3.1.4 Linguistic

- Generators of words (names, places, objects. . . ) which can use rules defined to infer words of a certain language.

### 3.1.5 Misc

- Determination of rules for species growth (weight, height. . . ).

### 3.1.6 Projects

- Management of the bibliography.
- Management of the source with *Version Control System* (VCS).
- Bugtracker for each book.

### 3.1.7 Community

- Wiki which reuses data of the other tools.
- News of the projects.
- Polls.
- Forum.
- Personal notes.

### 3.1.8 Writing

- Parse texts to make statistics, like showing repetitions.
- Helpers to translate texts.

### 3.1.9 External

- Use Gobby to collaborative writing.
- Let users chat with Jabber.

## 3.2 Fantasy/SF

### 3.2.1 Geodesic

Options are:

- calculate the smallest distance between to places in units defined by the user;
- display the time of the travel, depending the transport;
- calculate the angle between two places.

### 3.2.2 Growth

A length is linked to the age with an equation. Other are proportionals to this length.

The user enters several couple of points and the program try to determine the best equation. The equation is plotted to let the user verify it.

The user can also enter a function he have invented.

## 3.3 Linguistic

### 3.3.1 Generators

The generators let the users to generate random thematic words. For example :

- names;
- cities, places;
- planets
- objects;
- animals...

Users can use defined rules of languages they have invented. The can also use translators (from internet) to create words.

Algorithms:

- Soundex.

## 3.4 Projects

## 3.5 Writing



# USER DOCUMENTATION

## 4.1 Markup

Available markup are:

- Markdown (default);
- HTML.

Check the respective documentations to learn how to use them.

Some additional markup are implemented.

### 4.1.1 Wiki

Link to wiki pages are denoted with bracket `[[pagename]]`.



# CODE DOCUMENTATION

Contents:

## 5.1 core — Core modules

### 5.1.1 constants — Constants

This module define the main constants of the program, like paths, version...

**get\_version** (*release=False*)

Return the version in a readable form.

**Parameters** **release** (*boolean*) – if True, return also the release

**Returns** a string of the version

**APPNAME**

The name of the application.

**VERSION**

The version of the program.

**WEBSITE**

Website of the project.

**ROOT\_DIR**

Root directory of Azur files.

**RESSOURCES\_DIR**

Ressources directory for Azur: it contains all media, like images, used by Azur.

**TEMPLATES\_DIR**

This directory contains all the templates which are not associated with a specific module.

**STATIC\_DIR**

This directory contains all static medias used by the Django website, like Javascript code, CSS files, etc.

**WHOOSH\_INDEX\_DIR**

This directory is used by Whoosh to store its index.

### 5.1.2 core.markup — Constants

Constants and functions to use markups in text.

**make\_interpage\_links** (*content*)

Replace markup with interwiki page links.

**make\_toc** (*content*)

Generate a table of content for an html document and place it at the beginning.

It takes only h2 and lower titles.

**parse\_text** (*content*, *syntax='markdown'*, *interpage=True*, *toc=True*)

Parse text to transform markup into html.

**MARKUPS**

Available markups in text fields.

## 5.2 settings — Django settings

Configuration file for Azur project. Adapt it to your system.

**EVENTS\_MODELS**

This dictionary lists the different types of event models. Key and value are on the shape 'name'- 'module.model'.

**CHARACTERS\_PROFILE\_MODELS**

This dictionary lists all the extensions for character profile. Key and value are on the shape 'name'- 'module.model'.

**WIKI\_EXT\_OBJECTS**

This dictionary lists all models for which one wants to create virtual page in the wiki. Key and value are on the shape 'model'- 'field name for searching'.

**WIKI\_EXT\_APP**

This tuple lists all models for which one can create categories in the wiki. It contains 2-tuple ('model', 'name').

Azur uses several softwares:

- [Whoosh](#): this is a search engine coded in Python. It is used through the Haystack application.
- [Markdown 2](#) Python package.

You need to install them in order to make Azur working.

Some Django external applications are used (but they they are already installed in Azur):

- [Haystack](#): this application provides a powerfull search engine.



# GLOSSARY

**Bugtracker** A bugtracker is a tool in which users report issue occurring in the software. It can also be used to propose improvements.

**Forge** A forge is a platform to allow collaborative software development over internet. It aggregates a set of applications in this view.

**Version Control System** Version control is the management of changes to documents, programs, and other information stored as computer files.



# AUTHORS

- Harold Erbin <[harold.erbin@gmail.com](mailto:harold.erbin@gmail.com)>



# THANKS

- Stéphane Bernard (logo)



# COPYRIGHT

Azur and this documentation are :

Copyright © 2008–2010 Harold Erbin <[harold.erbin@gmail.com](mailto:harold.erbin@gmail.com)>

## 9.1 Licence

Azur is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation version 3 of the License.

Azur is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with Azur. If not, see <http://www.gnu.org/licenses/>.

**See Also:**

**COPYING** A copy of the whole licence text.





# CHANGES IN AZUR

## Table of contents

- Release 1.0 (in development)

## 10.1 Release 1.0 (in development)



# INDICES AND TABLES

- *genindex*
- *modindex*
- *search*
- *Glossary*



# PYTHON MODULE INDEX

## **C**

`core`, 12  
`core.constants`, 11  
`core.markup`, 11

## **S**

`settings`, 12



# INDEX

## A

APPNAME (in module core.constants), 11

## B

Bugtracker, 13

## C

CHARACTERS\_PROFILE\_MODELS (in module settings), 12

core (module), 12

core.constants (module), 11

core.markup (module), 11

## E

EVENTS\_MODELS (in module settings), 12

## F

Forge, 13

## G

get\_version() (in module core.constants), 11

## M

make\_interpage\_links() (in module core.markup), 11

make\_toc() (in module core.markup), 12

MARKUPS (in module core.markup), 12

## P

parse\_text() (in module core.markup), 12

## R

RESSOURCES\_DIR (in module core.constants), 11

ROOT\_DIR (in module core.constants), 11

## S

settings (module), 12

STATIC\_DIR (in module core.constants), 11

## T

TEMPLATES\_DIR (in module core.constants), 11

## V

VERSION (in module core.constants), 11

Version Control System, 13

## W

WEBSITE (in module core.constants), 11

WHOOSH\_INDEX\_DIR (in module core.constants), 11

WIKI\_EXT\_APP (in module settings), 12

WIKI\_EXT\_OBJECTS (in module settings), 12