WotWizard Manual

WotWizard builds, from the blockchain and sandbox of Duniter, a prediction of the future entries of candidates into the Duniter Web of Trust (WOT). It uses a simulation of the Duniter mechanism. When several possibilities may happen, each one is listed with its probability. The published lists are automatically updated every five minutes, and any change is signaled visually.
WotWizard contains several tools and a fast **Web of Trust Explorer**.

Warning: You must use the 1.7.17 version of duniter or any later version in the 1.7.x series. Versions 1.8.x don't work with WotWizard.

How to use it?

This program needs a **Duniter** node running on the same computer.

This node must be configured for the use of WotWizard. To do so, edit its configuration file ("conf.json"; see *Configuration* below) in a text editor and put the value of the "storage.wotwizard" field to "true".

WotWizard includes a server executable (**wwServer**) written in *Go* (v1.15.2) and running under GNU / Linux. This server communicates by the way of http POST methods containing *GraphQL* requests on input, and JSON answers on output. WotWizard includes an optional graphical user interface (**wwClient**) written in Go too and interacting with the server through http and with users through a web browser.

wwServer is to be placed in an empty directory where it can be run. It creates itself the sub-directories and files it needs, all of them being included in the inital directory. If the Duniter database is not at its standard place, you can give it with a command line option.

wwClient may be placed in its own empty directory or in the same as wwServer, where it creates its own resource files.

Server Mode

You can use WotWizard as a server only, if you want to display its results yourself, e.g. in a web page. Configure first Duniter for WotWizard (see above). If Duniter installation is standard, (its database must lie in the file "~/.config/duniter/duniter_default/wotwizard-export.db"), you can run wwServer without options:

```
$./wwServer
```

Otherwise, on the first run, add the -du option:

```
$./wwServer -du <path to Duniter database>
```

On later runs, the -du option is no more useful.

It's also possible to modify this path later by editing the "rsrc/duniter/init.txt" file. This change is taken into account only when wwServer (re)starts.

On the first run, after a not-so-long initialization phase (which starts after the recording of a new block in Duniter) that you can follow in the "rsrc/duniter/log.txt" file, request files can be sent. In the case of errors in requests, error messages appear in log.txt and in the "errors" field of the JSON answers.

By default, the server listens on "localhost:8080". To change this, edit the "rsrc/duniter/serverAddress.txt" file and (re)start wwServer.

To stop wwServer, press Ctrl-C or use any other standard method.

The WotWizard database is contained in the "rsrc/duniter/System/DBase.data" file. At every start of wwServer, a new copy of DBase.data is created. At most, two copies are saved: "Dbase.data.bak" is the last one and "DBase.data.bak1" is the previous one. If you have a doubt on the integrity of the datebase, replace it, once wwServer is stopped, by renaming an older copy.

GraphQL requests, and their meanings

The content of request files must follow the graphQL specifications of june 2018:

https://spec.graphql.org/June2018/

The graphQL types definitions document for WotWizard is available in the documented Help/TypeSystem.txt file.

The syntax of graphQL requests is described in the Help/graphQL_ebnf.txt file (look at *ExecutableDefinition*).

Requests are form requests with fields:

Field key	Field value	Needed?
graphQL	Text of the graphQL request	Always
opName	Name of the requested *operation*	If more than one operation in the graphQL request
varVals	graphQL variables values (JSON text)	If the requested operation contains variables
returnAddr	Url where a *subscription* response is sent	If the requested operation is a subscription

Interactive Mode

Place the "wwClient" file in the same directory as wwServer or in another empty file. Run wwClient without options:

```
$./wwClient
```

By default, the server listens on "localhost:7070". To change this, edit the "rsrc/duniterClient/htmlAddress.txt" file.

Verify in the "rsrc/duniterClient/serverAddress.txt" file that the address where the client send requests is the same as in the "rsrc/duniter/serverAddress.txt" file (by default "localhost:8080").

By default, the return address for subscriptions is "localhost:9090". To change this,

edit the "rsrc/duniterClient/subAddress.txt" file.

After editions, (re)start wwClient (once wwServer has notified in the terminal it's listening).

Once wwClient has started, choose your language (last link in the index).

After that, you can select the links you want.

Note: In this version, graphics are not implemented yet. Just lists of values are displayed.

Forecasts

WotWizard View gives the list of the future entries in the Web of Trust (WoT).

You can:
- choose the way the list is displayed (by names or by dates, or metadata -- analog to the 'Preparatory File' data: see below);
- manually update the list.

The number of permutations is the number of different possible orders of entries.

Preparatory File: WotWizard calculates its forecasts from a state of affairs called *File*. It's a list of *dossiers* (newcomer alias + certifications list) and *internal certifications* (between members), sorted by availability dates. Its structure is:

- *Internal certification* : one line with:
  + the receiver and sender of the certification (aliases) with an arrow (←) between them;
  + the certification availability date;
  + between brackets, the validity limit of the certification;
- *Dossier* :
  + a first line describing the newcomer:
    * the number of main certifications (those which fix the availability date of the dossier, see below);
    * her alias;
    * between brackets:
      - her availability date;
      - its minimal availability date, two months (sigReplay) after her previous membership application;
    * between brackets, the validity limit of her application;
    * the distance rule result, as the percentage of reachable sentries;

* a summary of the dossier state: OK if it's valid and if the rule of minimal certification number and the distance rule are verified, or KO otherwise (certification availabilities are not checked here).

+ one additional line for each external certification, in the same form as for internal certifications, but without the sender alias (which is the newcomer's); these certifications are sorted by availability date.

Main certifications: In a Dossier, it's the *n* first external certifications (in the order of availability dates), where *n* is the number of certifications it owns when it is less than or equal to *sigQty* (5). If it is greater, it's the number of certifications needed to satisfy the distance rule, or the total number if the distance rule can't be satisfied.

Permutations : displays all entry ranks permutations predicted by WotWizard, with their probabilities. **Warning**: their number may be sometimes very big and the display very large!

Web of Trust Explorer

You can search any identity in the blockchain or the sandbox by typing its first characters or the first characters of its public key into the first top field and by clicking on the "OK" button. The possible identities appear in the list below: choose the one you want to look at, and click again on the "OK" button. Several informations on the identity appear.

Identities can be partitioned in four classes, labeled by marks:

- Members (no mark)

- Newcomers (°), whose identities are in the sandbox and who are to enter the Wot.

- Excluded or missing (×), who were but are no more members for two possible reasons:

+ membership not renewed after one year

+ loss of their fifth certification (each certification has a validity delay of two years)

When excluded, an old member must apply to a new membership and get at least fifth valid applications before the second anniversary of its last membership application; otherwise it is revoked.

- Revoked (¶), who can no more enter the Wot with the same nickname and / or the same public key.

Checkboxes allow each class to be added or not to the list, or operations with costly calculation times to be performed or not.

If a member's end of membership has been requested, her expiration date of membership is preceded by a cross (×). In this case, she can no more get new certifications, and, without renewal of her membership, she will be excluded at this date.

Properties

Distances: Give the distances of all members to the web of trust (proportion of sentries reachable in at most *stepMax* (5) steps) sorted by proportions (with drawing) and by aliases.

Qualities: Give the qualities of all members sorted by qualities (with drawing) and by aliases.

Centralities: Give the centrality levels of all members sorted by centrality levels (with drawing) and by aliases.

The degree of centrality c of a member is the number of oriented paths (certifier -> certified) of shortest lengths to which she belongs and whose length is, at most, *stepMax* (5) plus one steps from the certifier. The centrality level c' of a member is calculated from her degree of centrality c by the expression:

$$c' = 100 \frac{\ln(1+c)}{\max[\ln(1+c)]}$$

where $\max[\ln(1+c)]$ is the greatest value of $\ln(1+c)$ for all members, and where ln is the natural logarithm.

One may consider the *centrality level* of a member as the level of her involvement to help people she certified to respect the distance rule, independently of her capacity to do so. By opposition, the *quality* of a member is her capacity to help people she certifies to respect the distance rule, independently of her level of involvement.
A member with a quality greater than or equal to *xpercent* (= 80%), makes, only by herself, people she certifies respect the distance rule.

Limits

Limit Dates of Memberships: Display the validity limits of memberships for all members, sorted by dates.

Limit Dates of not-renewed memberships: Display the limit dates before revocation of identities with not-renewed memberships, sorted by dates.

Limit Dates of Certifications: Display the validity limits, for lack of certifications, of memberships for all members, sorted by dates.

Evolution

Number of Members: Give the list of block dates where the number of members changed and the corresponding members' number since the beginning of the money (with drawing).

Flux of Members: Changes in the number of members by time unit (i.e.: one month). So, it's the derivative of the number n of members with respect to time t : $\frac{\mathrm{d}n}{\mathrm{d}t}$ (with drawing).

Flux of Members per Member: Flux of members divided by the number n of members: $\frac{\mathrm{d}n}{n\,\mathrm{d}t} = \frac{\mathrm{d}\ln(n)}{\mathrm{d}t}$ (unit: percent by time unit). With drawing.

First Entries: Give the list of block dates where the number of members entering the WoT for the first time changed and the corresponding number e of first entries since the beginning of the money (with drawing).

Flux of First Entries: Changes in the number of first entries by time unit (i.e.: one month). So, it's the derivative of the number e of first entries with respect to time t : $\frac{\mathrm{d}e}{\mathrm{d}t}$ (with drawing).

Flux of First Entries per Member: Flux of first entries divided by the number n of members: $\frac{\mathrm{d}e}{n\,\mathrm{d}t}$ (unit: percent by time unit). With drawing.

Losses: Give the list of block dates where the number of members changed and the corresponding differences $l = e - n$ between the number e of first entries and the number n of members since the beginning of the money (with drawing).

Flux of Losses: Changes in the losses by time unit (i.e.: one month). So, it's the derivative of the losses l with respect to time t : $\frac{\mathrm{d}l}{\mathrm{d}t}$ (with drawing).

Flux of Losses per Member: Flux of losses divided by the number n of members: $\frac{\mathrm{d}l}{n\,\mathrm{d}t}$ (unit: percent by time unit). With drawing.

Informations

Blockchain Parameters: Display the basic parameters of money.

List of Revoked Identities: Display revoked identities with their aliases, public keys and hashes, and the dates of their entries.

List of Excluded Identities: Display identities whose memberships were'nt renewed, with their aliases, public keys and hashes, the dates of their entries and their limit dates before revocation.

Members List: Display identities of all members with their aliases, public keys and hashes, the dates of their entries and their limits of validity.

List of Newcomers: Display identities of all newcomers with their aliases, public keys and hashes, and the limits of validity of their applications.

Certifications From Members: Display all certifications in the blockchain, sorted by senders, with their inscription dates.

Certifications To Members: Display all certifications in the blockchain, sorted by receivers, with their inscription dates.

List of Sentries: Display the identities of sentries.

Use it and enjoy! - ¡Úsalos y disfrútalos! - Bonne utilisation - Приятного использованияп - Powodzenia - Viel Spaß

Gérard Meunier