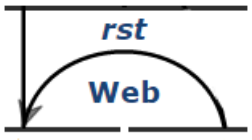


rstWeb
User Guide – Version 4.0.0

title:	rstWeb User Guide
rstWeb version:	4.0.0
guide version:	4.0.0.0
date:	2024-07-15

author:	Amir Zeldes
e-mail:	amir.zeldes@georgetown.edu
homepage:	https://gucorpling.org/rstweb/info/

Contents

Introduction

rstWeb is an open source, browser based annotation tool for discourse analyses in Rhetorical Structure Theory, and its enhanced version, eRST. It is meant to support collaborative, online annotation projects using just a Web browser, without installing software for annotators, though there is also a standalone local version for offline use if you do not have access to a server.

Getting Started

Local Installation

rstWeb runs in your browser no matter what, but you can run a local version of the software the emulates a Web server on your own machine. Getting this to work is a little different in Windows and Mac/Linux, and primarily requires Python and the library cherrypy to be installed (for Linux, replace your package repository for the Mac's easy_install, i.e. apt-get, yum etc. depending on your Linux flavor) If you want to use screenshots in the interface, you will also need to install **Selenium**, which works with PhantomJS to create screenshots (see below).

1. Make sure Python is installed (2.6 or newer; up to 3.11 has been tested):
    - For **Mac**, Python is typically installed by default, no need to do anything
    - For **Windows**, download and install Python from https://www.python.org/
2. The Python packages *cherrypy* and *selenium* must be installed if they aren't already: (selenium is only required for exporting screenshots, you can skip it if not needed)
    - For **Mac**, you may need to install pip first, by opening a terminal and typing:
      sudo easy_install pip
      Enter your password, and once pip is installed, run these commands:
      sudo pip install cherrypy
      sudo pip install selenium
    - On **Windows**, pip is installed by default with Python, so you should only need to open a command line (Start menu -> run -> cmd) and type:
      pip install cherrypy
      pip install selenium
      Tip: If 'pip' does not work, you can also try: *python -m pip install cherrypy*
3. Unpack all of the files from the rstWeb repository in Github to some folder
4. Run the appropriate script:
    - On **Mac/Linux**: run *rstweb_local.sh*
    - On **Windows**: run *rstweb_local.bat*
5. You can now use rstWeb in your browser at: http://127.0.0.1:8080/

If you run into problems getting the software to run, please contact amir.zeldes@georgetown.edu

Server Installation

1. Make sure Python is installed (preferably 2.6 or newer, of 3.5 or newer for Python 3)

2. Unpack all of the files from rstWeb to the directory they will be served from
3. Configure your Web server to have read, write and execute privileges within this folder
4. Make sure selenium is installed (e.g. via pip), otherwise screenshots won't work
5. You may want to forbid users from interacting with files other than the top level python scripts in the main rstWeb directory (in particular, no one should have access to the configuration files)
6. You will also want to disallow or simply delete the local version's launch script, start_local.py, since server users shouldn't be able to use it. You may also remove the .bat and .sh scripts.
7. If you're using Apache, here is a possible configuration file:

```
Alias "/rstweb" "/var/www/html/rstweb"
<Directory "/var/www/html/rstweb/">
RewriteEngine On
RewriteBase /
DirectoryIndex open.py

<IfModule mime_module>
AddType application/x-httpd-py .py
</IfModule>

RedirectMatch 404 ".+\.(py(c|o)|db|txt|rs3|ini)$"
RedirectMatch 404 ".*/(modules|export|import|templates|users).*$"

AddType text/html *
Options Indexes FollowSymLinks MultiViews
Options +FollowSymLinks
Options +ExecCGI
AddHandler cgi-script .py
AllowOverride None
Order allow,deny
allow from all
</Directory>
```

8. Use the administration interface to change the passwords/user names to secure your system. The initial administrator password is 'pass1'.
9. You're all set!

Usage with Docker

This option is primarily useful for sys admins automating deployment of rstWeb (or multiple instances of rstWeb), if you are running an operating system supporting docker. You can install rstWeb in a docker container using the following steps.

Note that current docker releases do not support **RHEL 6** and that there are some known conflicts if you are running **SELinux**. Under those circumstances running over the normal apache server is recommended.

1. Build docker container.

```
cd docker
./build.sh
```

2. Run docker container. The default port in the script is 8085

```
cd ../
./start_docker.sh
```

3. You can now use rstWeb in your browser at: http://127.0.0.1:8085/rstweb/open.py

Troubleshooting

If you're having trouble, it's possible some permissions are set incorrectly, or that your server needs to be configured to execute the Python scripts. Otherwise, the entry point for the program is the script *open.py*. If you're using the Apache configuration above, this acts as the directory index, so you can simply direct users to http://.../<rstwebsdirectory>/.

Upgrading: If you have upgraded from an older version of rstWeb, it is recommended to click on admin -> database -> update schema (NOT 'init DB'). This should not result in data loss, but backing up the rstweb.db is always a good idea. If new features are not working, you should also empty your browser cache to reload all css and javascript updates.

Logging in

Local version

If you're using the local version, you will be automatically logged in as the user 'local'. You will have administrator rights (see Administration), but User Management will be disabled (the user 'local' will be used for all annotations.

Server version

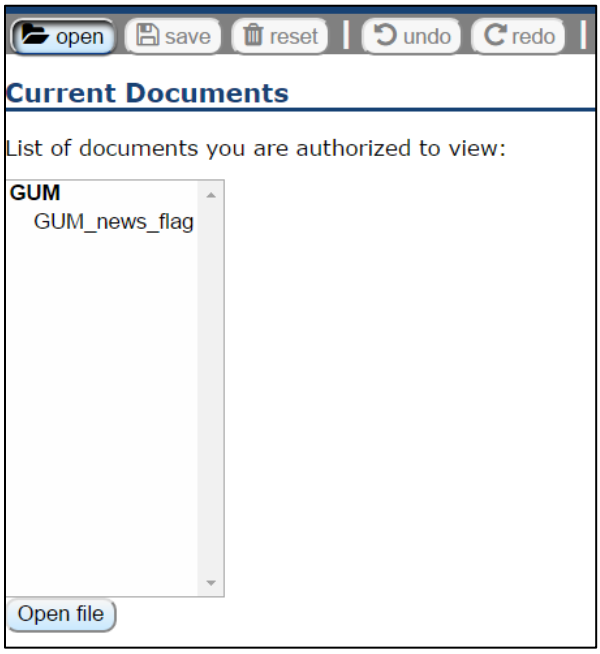
If you are working on a server over the internet, a login will be provided for you by an administrator (see User Management below). To log in, go to the server URL, e.g. https://gucorpling.org/rstweb/ - if you are not already logged in, you will be prompted to enter your credentials. To create new user names as an administrator, see Administration.

Annotation

Opening a document

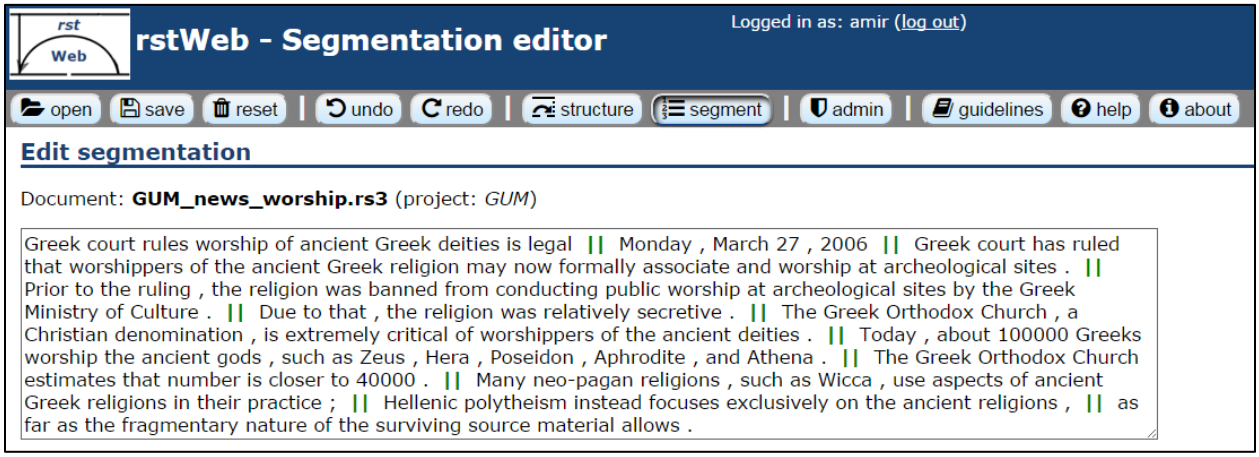
Once you are logged in, you will see a list of the documents you are allowed to edit, grouped by annotation project. You can return to this screen using the *open* button.

Administrator users may see all documents. Note that every user has their own copy of the documents they are working on, so that multiple annotated versions of each document can exist simultaneously.



There are two modes while annotating: **segmentation mode** and **edit structure mode**.

Segmenting units



In segmentation mode, you can split the text into Elementary Discourse Units (EDUs). Initial EDU borders can be set by either importing an already segmented .rs3 file (e.g. from RSTTool), or by importing a plain text file with one EDU per line (see Import under Administration). You can alter segmentations at any time by pressing the *segment* button.

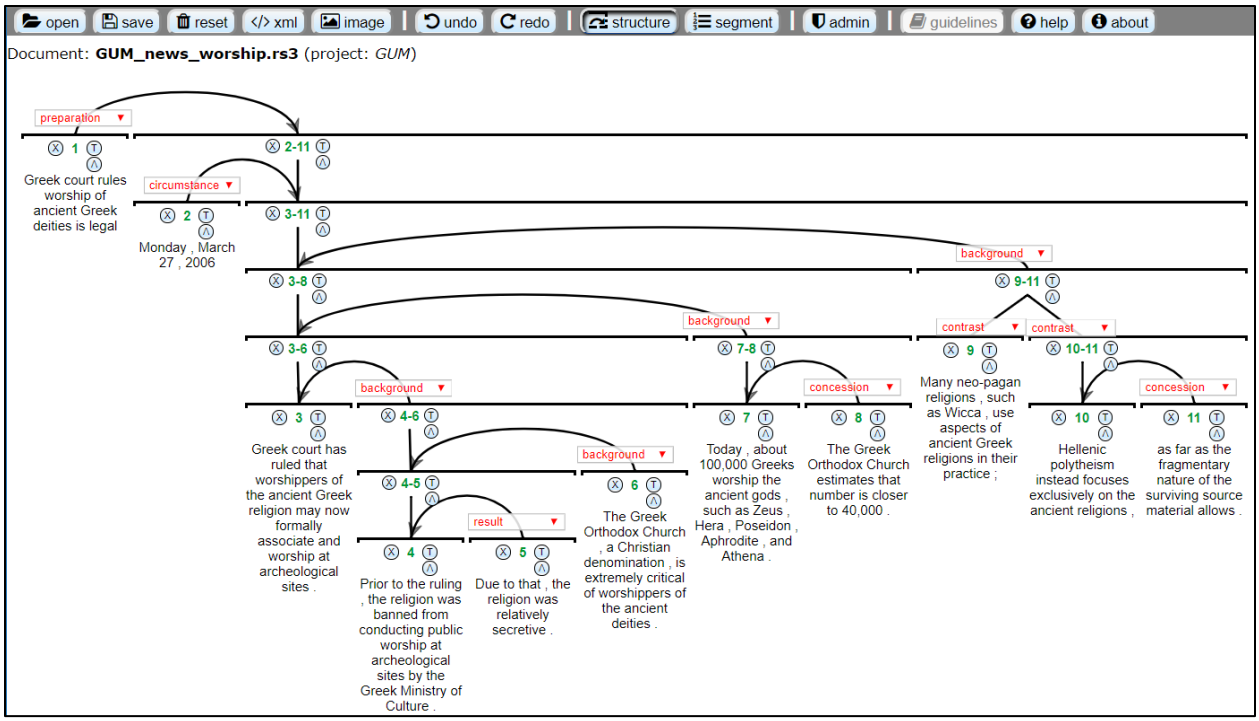
To add segments, click on a space between words. To remove segments, hover over the green dividers between segments and click on the x that appears to remove them.

It is possible to add and remove segmentations of already structured texts. In this case, adding a segment will cause the first part of that segment to remain linked to the annotation graph, and the second part will be inserted as a new unattached EDU for annotation. Deleting a segmentation border will merge the second unit into the first, so that the relations involving the first part of the new merged segment are retained, but those for the second part are deleted.

If a guideline link has been set for the current project, the *guidelines* button will allow you to access guidelines for your project (see Administration).

Editing structure

To annotate RST or eRST relations use the *edit* button. The interface below shows an annotated document with relations already in place.

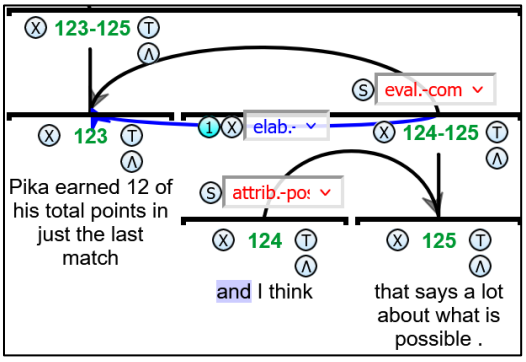


In edit mode, each elementary discourse unit (EDU) has its text under a line with a number giving its position in the text. Similarly, groups of units have a range of numbers. You can

connect nodes, unlink them, and group them together under spans and multinuclear nodes. You can undo and redo actions at any point using the appropriate buttons.

Connections

- To connect nodes, drag from the numbers under one unit to the numbers under a target unit
- To change the relation between two units, use the drop down list on the connection between them
- To unlink a node from the graph, click its ⓧ button - all of the nodes connected above it will be unlinked
- **eRST:** to create an eRST secondary edge between nodes, hold **ctrl** while dragging/dropping. These edges appear in a different color, and do not re-arrange the primary, as shown in the figure below.



Spans

- Use the Ⓣ button to add a span above a node. The span will group together all of the nodes connected to that node

Multinuclear relations

- Use the Ⓐ button to create a multinuclear node (multinuc) above a node. The multinuc can have multiple child nodes with the same relation (e.g. a set of items in contrast to each other)
- To change the relation of a multinuc to its children, change the relation box of any of its children
- When connecting a new node to a multinuc node, it will be added as a multinuc child by default. You can change the relation to a satellite relation by choosing "change to satellite" in the relation box

Don't forget to save your work! If you make a mistake and want to go back to the original form of the document when it was imported, you can also use the *reset* button (**warning:** this will delete all of your annotations for this document!)

If a guideline link has been set for the current project, the *guidelines* button will allow you to access guidelines for your project (see Administration).

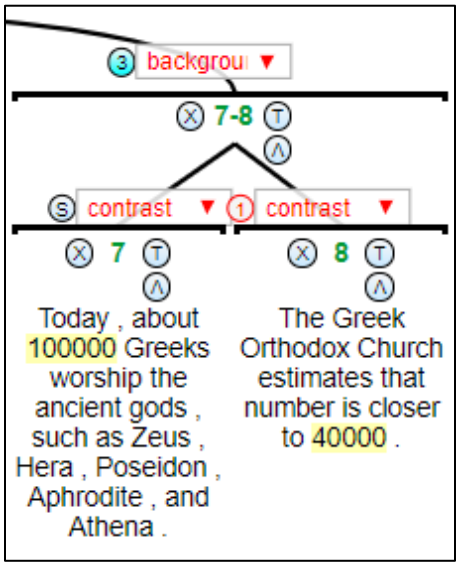
The XML button can give you a quick export of the current document in the .rs3 format, and the image button will download an image of the last save version of the graph (only if Selenium is installed, see Installation).

Adding signals

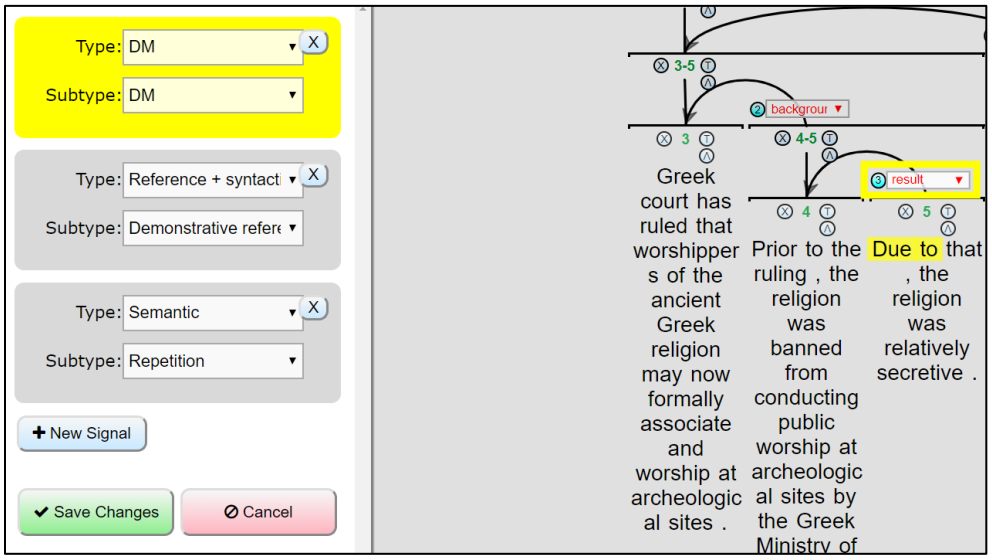
rstWeb allows users to add information about the presence of signals which allow analysts to identify discourse relations (for example connectives, such as 'however', indicating contrast, or groups of word such as 'as a result'). Signals can be associated with any number of words (including no words, e.g. if the signal is a 'genre convention' or 'graphical layout'), a major category (e.g. 'morphological') and a minor category (e.g. 'tense'). The same words may be associated with multiple signals.

Access to the signaling annotation functionality can be turned on/off in the Administration interface, in the **Database** tab. To add signals to a document, a signaling annotation scheme must be defined in a .json file in the directory signals/. You can find an example in **signals/default.json**. If you have multiple signaling schemes, you can change the active scheme in the Database tab as well.

Once a scheme is defined and signaling annotation is turned on, you can import a document and begin annotating signals using the Ⓢ button next to the relation chooser in editor mode. If a relation already has some signals, the button will be colored blue and show the number of existing signals. Signal words will be highlighted when the button is hovered over.



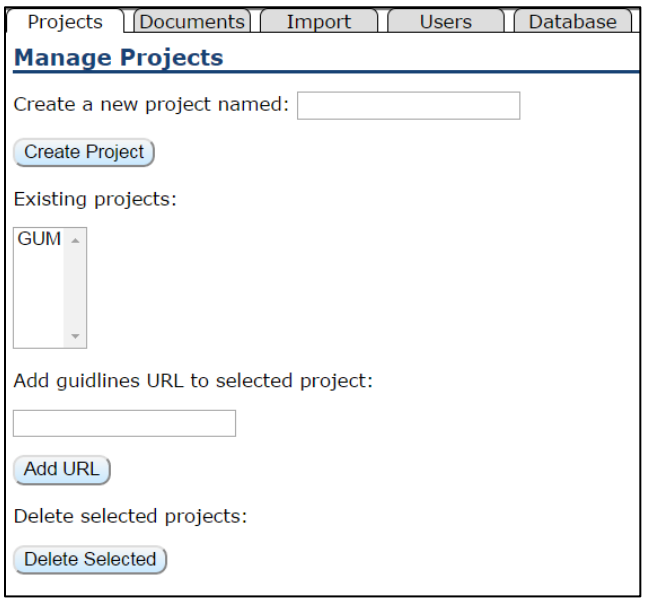
Clicking on the button will open the signal annotation tray, allowing users to choose a signal type and subtype, and click on associated tokens to highlight or un-highlight them.



Administration

The administration interface is only available to administrator users (marked with level 3 in the user file in the users/ directory; the default password for admin is ‘pass1’, and this user can create new administrators). The interface contains five tabs:

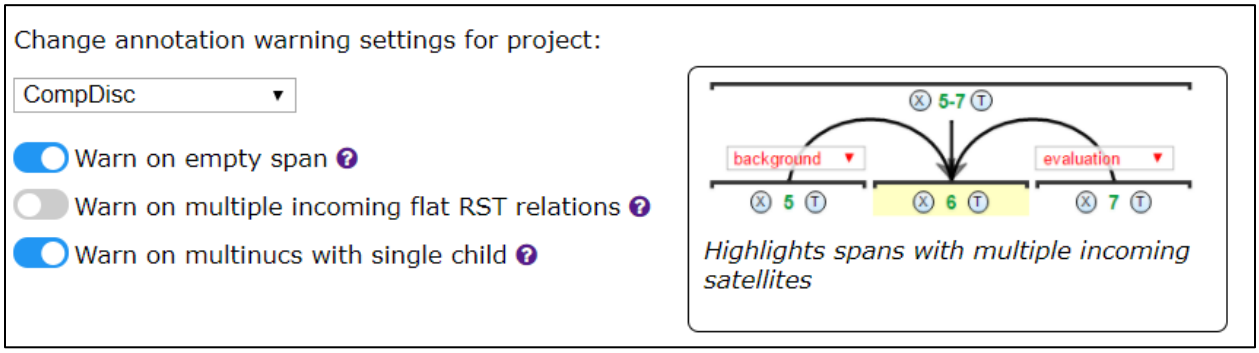
Projects



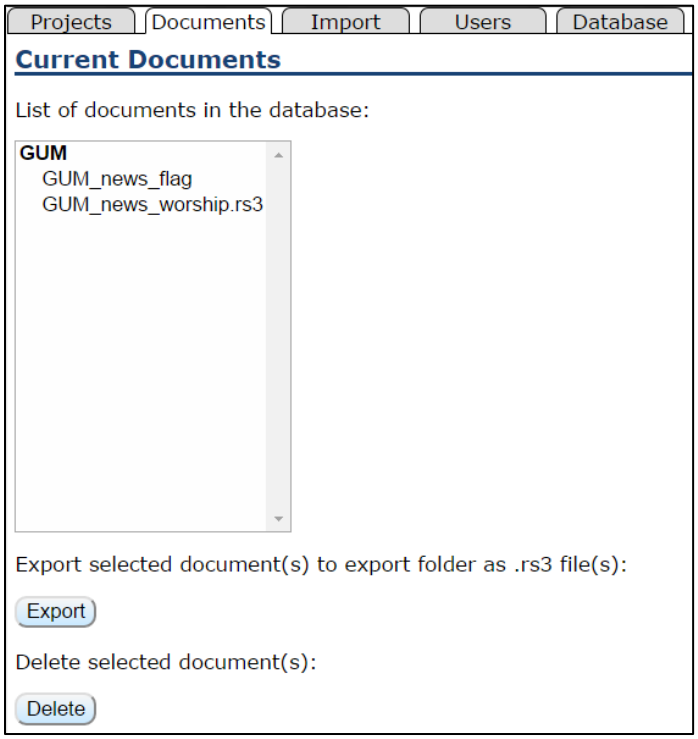
This tab lets you create new projects and delete existing ones. Projects are used to group documents together. Deleting a project will remove all documents within it permanently!

You can also add a URL giving guidelines for one or more projects. If you enter a URL and select multiple projects, all of those projects will be given the guideline URL, which is available from the *guidelines* button while editing. To change the URL, simply specify the new one and click *Add URL*.

If you wish to show warnings with yellow highlighting when users create multinucs with only one child, or empty hierarchy spans, or multiple incoming RST relations into the same node, you can turn on these inspections per project in this tab using these toggles:

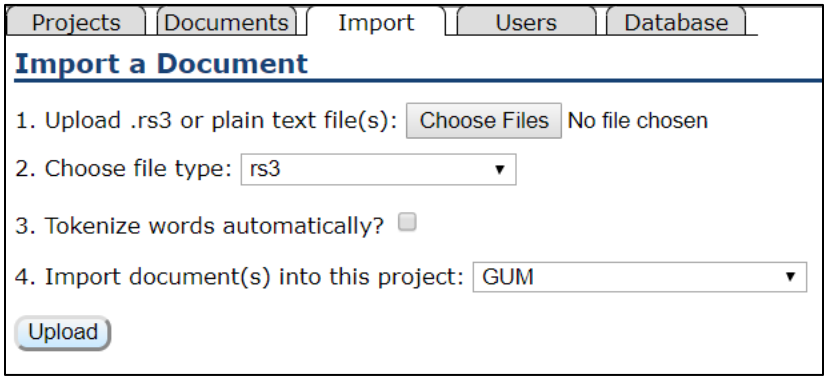


Documents



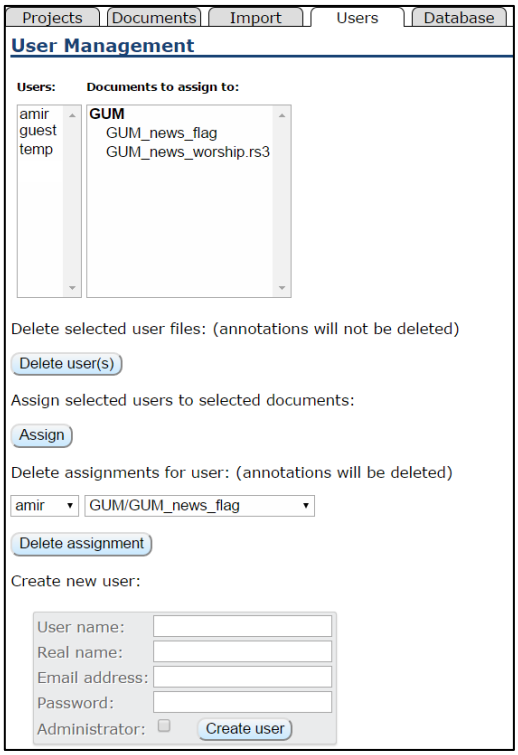
This tab shows you a list of all documents and allows you to delete them or export existing document annotations. If multiple annotators have worked on a document, one rs3 file will be created for each annotator's version in the export/ folder.

Import



Here you can import new documents. Choose between .rs3 format (from RSTTool) and plain text (one EDU per line). When importing plain text files, the default relations that are available for annotation are determined by the file *default_rels.tab* in the *users/* configuration directory (alternatively you can choose a different location in *users/config.ini*).

By default, rstWeb will assume that whitespace separates individual tokens within discourse units. This is mainly important if you want to use discourse signal annotations, which are attached to individual words. If your text is not whitespace tokenized, you can check "Tokenize words automatically", which uses a basic tokenizer which reasonably for European languages. If you want more control over tokenization or are working on signal annotations in a language without spaces, you can pre-tokenize your data so that words have spaces between them.



The new documents will be imported into a new project and a version of the document will be created for your user. You can assign the document to other users in the Users tab. A copy of the imported document will be saved in the import/ folder and you can always revert to the imported state of the file while editing by using the *reset* button after you've opened the document.

Users

The Users tab lets you create and delete user files.

These are stored as .ini files in the users/ directory. User management is only possible in server mode – this tab will be disabled in local mode and the default user will always be 'local'.

New users can be defined as either normal users or administrators using the checkbox in the new user form at the bottom. Administrators can see all documents and have access to the administration interface. New user names cannot be: **'config', 'default', 'local', 'temp', 'emails', '_orig'** or **'pending'**.

For normal users, you need to assign a user name to a document name in order to make that document visible to the annotator. You can assign multiple annotators to multiple documents simultaneously by selecting multiple lines in the Users and Documents to assign boxes.

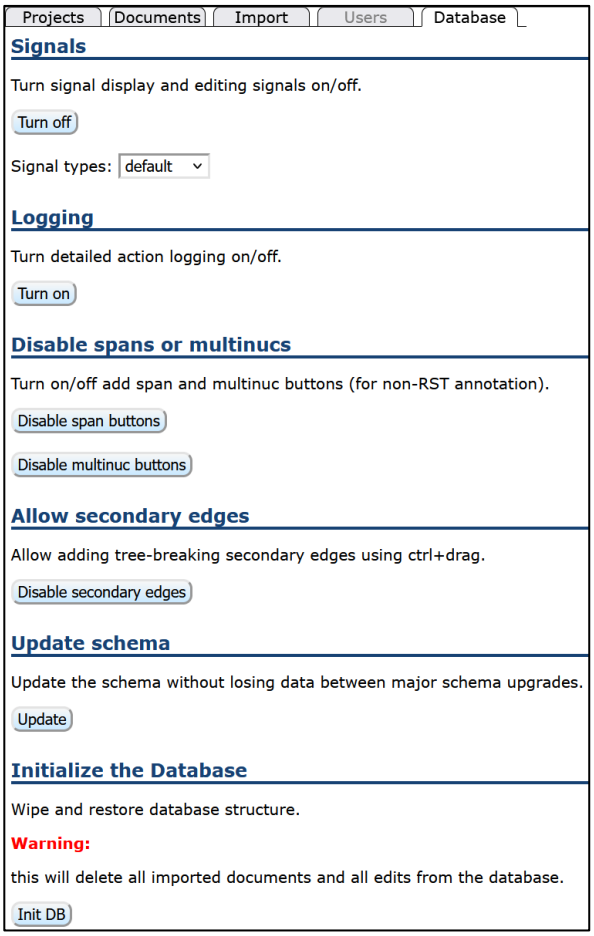
Once you assign a document, a copy of it will be created in the system which can then be annotated and exported without affecting other copies of that document.

Deleting an assignment will delete that user's version of the document, but will not affect other users' annotations and will not remove the document from the system. To completely remove a document, use the delete function from the Documents tab.

Database

This tab allows you to enable/disable signaling annotation, turn detailed logging on or off, and update the schema from older versions of rstWeb to the latest version of the software (this should usually not result in loss of data, but make a backup of your database just in case). The logging functionality records all submitted editing operations (after clicking *save* in the editor), allowing you to retrace the entire annotation history of each document, when logging is on.

It's also possible to disable the use of the add span/multinuc buttons in the interface. This is potentially useful if you want to annotate outside of the framework of RST (e.g. binary discourse relations), and want to prevent users from adding multinucs or spans, if your annotation framework does not support those.



The button “Disable secondary edges” determines whether **eRST**-style secondary edges can be superimposed on the basic RST tree by holding **ctrl** while dragging a connection.

Finally, you can also wipe the database clean and restore its original schema, current to the installed version. This is mainly useful if the database file becomes corrupt somehow, or if you want to quickly delete all data in the database (delete all documents and projects).

Configuration file

Some aspects of the interface, especially file and directory paths for templates, default relations etc. are configured in *users/config.ini*. You can edit this file to use different templates, place import and export files in different locations, and in the future also to use automatic mail notifications to new users (not yet supported).