Reverse-Engineering Database – An IDA-Pro Plug-in

Server setup instructions

1. Assumptions:
    a. The server was developed on Windows 7 and on Ubuntu 10.04.
    b. We used Django 1.4 for the server's functionality. We used the web server included in Django for the development. Refer to Django's site for information regarding the usage of other web servers.
    c. We used SQLite as a database. Refer to Django's site for information regarding the usage of other databases.
    d. Django requires Python to run. We used 2.6 and 2.7 during the development.
2. Instructions:
    a. Install NetworkX on the machine. Version 1.7 was user during the development.
    b. Download the server 7z file.
    c. Extract it to any location on your machine.
    d. Configuration:

    Django has a lot of configuration options. We won't cover all of them here. We will only note the ones necessary specifically relevant to REDB. For further information regarding the setup of a Django server, refer to Django's site.

        i. The "function_description_db" Python module (directory) should be alongside the manage.py file.
        ii. In "server\server\settings.py" the default DB must be set (This is where the Functions and the Descriptions will be saved). Also add "function_description_db" to INSTALLED_APPS.
        iii. "server\server\urls.py" should contain the following lines:
            1. url(r'^request/', 'function_description_db.views.request_handler')
            2. url(r'^submit/', 'function_description_db.views.submit_handler')
            3. url(r'^compare/', 'function_description_db.views.compare_handler')

            The third one is needed only if you wish to allow custom queries. See explanation on the REDB Lab module in the site.

    e. In order to create a new DB run the following command on CMD/BASH when the current working directory the server's directory (when you can see manage.py using dir/ls):

    python manage.py syncdb

    f. Run the server using the following command in the same directory:

Python manage.py runserver <IP>:<Port>

(Set IP and Port to the wanted values. By default, when no values are entered, the server will run locally on port 8000.)

g. If all goes well a message saying the server in running will be printed. Otherwise, an error message will appear stating the nature of the problem. Refer to Django's site, and specifically the tutorial.

If you encounter problems, please contact us using the contact information provided in the REDB site.