

Sydney Street Art - documentation

Kamil Korzekwa*

I. DATA

All pictures of the graffitis are stored in **photos** folder within subfolders named after the suburb the graffiti is located in. The file name pattern is **suburb_name+location_number+image_number**, i.e. pictures with very similar GPS coordinates have the same location number and differ only by the image number (on the map only one marker is shown and all pictures with a given location number correspond to it).

The information about all graffitis is stored in **data.xml** file within **data** folder. Each graffiti is a separate **xml** element **<graffiti>** with information encoded in the values of its subelements (fields), e.g. **<file_name>** or **<artist>**. The information about graffitis can be divided into two categories:

1. *Automatic*: Information about graffitis that can be automatically retrieved by the software from **.jpg** files (e.g. GPS coordinates) and the folder structure (e.g. suburb name if graffitis are properly grouped within suburb subfolders).
2. *Manual*: Information about graffitis that requires manual input (e.g. artist's name).

Tools designed to deal with automatic information are stored in **software** folder, and the ones to deal with manual information in the **data** folder.

Besides **data.xml** the information is also stored in folders **artists** and **suburbs**. Each of the folder contains two files per suburb/artist: **name.xml** and **name.description.xml**. The first one is just a smaller copy of **data.xml** containing only graffitis relevant for the given artist/suburb. The second one contains longer descriptive information about the artist/suburb.

A. Automatic information

Tools used to deal with automatic data are located in the folder **software** and are written using **python**. The central program is **extract_data.py**. It scans over all **.jpg** files within a given folder (**photos** by default) and extracts the following information: file name, full path to the picture, suburb (from the name of the subfolder within **photos** folder), date and time the photo was taken, GPS coordinates, height and width of the photo. It then saves the information to **data.xml**, also adding the field **<photographer>** (by default set to 'KK', but can be changed each time a new batch of files from a given photographer is added) and empty fields: status, title, artist, link to artist's website and description. These fields need to be modified manually. Each time the program is run, before proceeding it makes a copy of the current data in the **archive** subfolder.

There are also three small auxiliary programs to quickly modify all the entries in the database:

- **modify_data_add_field.py**: adds a new field to each **<graffiti>** element (if one wants to store new kinds of information).
- **modify_data_add_id.py**: adds id to every field of each element according to a pattern **file_name+field_name**.
- **modify_data_change_names.py**: changes the file names of all **<graffiti>** element according to a pattern **suburb_name+location_number+image_number**.

These programs can be easily modified to serve new purposes.

There are also two programs that create new smaller databases based on **data.xml**, but which contain only information about graffitis relevant for a given artist/suburb:

- **data_artist.py**
- **data_suburb.py**

Finally, there is a file **summary.py** that creates a file **summary.txt** with information about the number of graffitis for each artist/suburb.

* korzekwa.kamil@gmail.com

B. Manual information

The remaining information has to be entered manually. This includes status (meant to be left empty, or set to 'Removed' when the graffiti no longer exists, so on the website it will be rendered in sepia mode), artist and link to artist's website. If multiple artists created a given graffiti then their names should be separated by '/' symbol. Also, for every artist added in that way one has to manually create a new field `<artist_link_n>`, where `n` goes from 2 to the total number of artists added. The link fields may be left empty, but they necessarily must exist. This is the part that has to be done on the level of the text editor, as the `xsl` web browser tool described below does not allow for this (yet).

One could enter the information using a simple tool like Notepad++. However, it is very inconvenient to enter the missing data while not seeing the graffiti picture. Therefore, a special stylesheet file `data.xsl` was written that allows one to display `data.xml` as a webpage, i.e. all information about all graffiti is displayed in the web browser alongside their photos. One can sort the graffiti according to artist/suburb/etc. by modifying one line of code in `data.xsl` (the place is indicated in the comments). Moreover, the information is displayed in the form of input lines, so that it can be modified. After modifying the data one has to click the 'Submit' button (once for each modified graffiti), which runs a PHP script that modifies the file `data.xml`.

Since the input and modification of the files is done using PHP, one needs to run it on a server. The easiest way is to use Apache software within XAMPP package that imitates a server on an off-line machine. More precisely, one has to first install it, then set the root folder to be the same as the `data` subfolder of the website folder, and finally write `localhost` in the browser address bar. The only non-trivial thing is setting the root folder: one needs to go to `xampp/apache/conf` folder and open `httpd.conf` file. Then, inside this file, one has to change the path `xampp/htdocs` to the website's path.

II. WEBPAGE

A. General structure

1. List of scripts

B. Google maps

C. XML database

D. Responsive screen