EBAS File Index Readme

EBAS bulk extracts (archives of NASA-Ames files) from the EBAS database are accompanied by a file index database.

The purpose of the file index database is to help data users to find files in the archive. The file naming convention for EBAS NASA-Ames data files provides metadata to a certain degree, but not all metadata can be encoded in the datafiles' names. Problem cases are related to multi-variable files containing variables with different component names, matrices or instrument types.

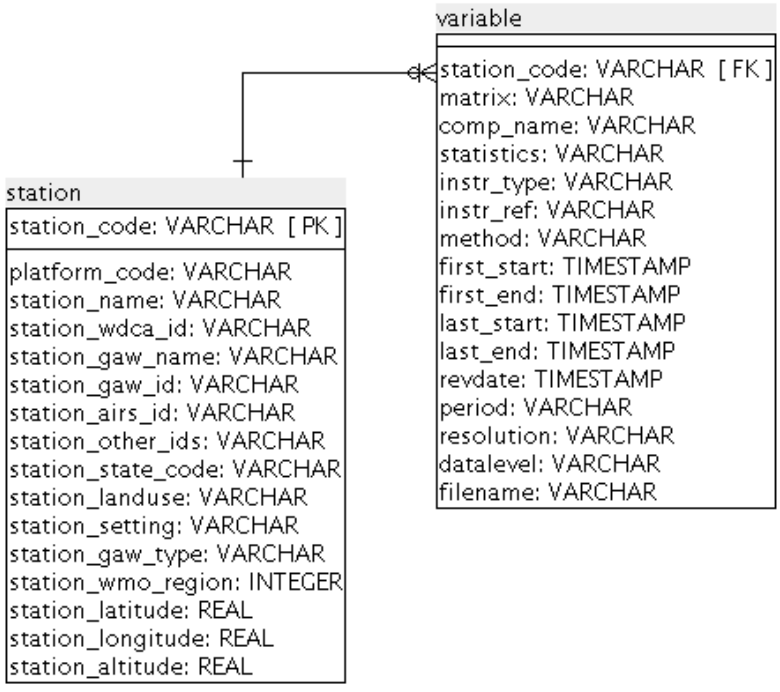
The file index database aims to fill this gap and enables data users to find exactly the files in the archive which contain the data variables of interest.

Database technology

The database is implemented as a sqlite3 database file. Sqlite is versatile, portable and easy to use on all platforms, and yet provides a rich set of relational database features.

Database layout

The file index database contains a table with station metadata (called *station*), and a table containing metadata for each variable in each file of the archive (this table is called *variable*).



The table layout can also be seen by issuing one of those commands in the sqlite commandline shell:

```
sqlite> .schema variable
```
or
```
sqlite> PRAGMA table_info(variable);
```

Quick intro – How to use the sqlite database

For users not familiar with sqlite, a good introduction can be found at the sqlite home page: http://www.sqlite.org (of particular interest might be the links to Getting started and Command Line Shell For SQLite).

Using the database

The easiest way to use the file index database is by using the sqlite commandline shell and selecting the metadata using the sql query language interactively.

Another possibility is to use programs or scripts to access the database. There are interfaces for a large variety of languages, e.g.: C/C++, python, perl, R, Matlab.

SQL query language

Learning SQL from scratch for the sole purpose of using the file index database might seem too much effort. While knowing SQL is definitely an advantage, using the examples below and adapting the examples to one's needs seems a perfectly suitable way.

Examples

Example 1: Open the file index database using the sqlite command line shell:

```
$ sqlite3 ebas_file_index.sqlite3
-- Loading resources from /home/nobody/.sqliterc
SQLite version 3.8.2 2013-12-06 14:53:30
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite>
```

Example 2: Select all station metadata from all stations (included in the archive), sort the output by station code:

```
sqlite> select * from station order by station_code;
station_code  platform_code  station_name  station_wdca_id  station_gaw_name  station_gaw_id  station_airs_id  station_other_ids
station_state_code  station_landuse  station_setting  station_gaw_type  station_wmo_region  station_latitude  station_longitude
station_altitude
------------  -------------  ------------  ---------------  ----------------  --------------  ---------------  -----------------
------------------  ---------------  ---------------  ----------------  ------------------  ----------------  -----------------
----------------
AM0001R       AM0001S        Amberd
40.38444444       44.260583333       2080.0
AT0002R       AT0002S        Illmitz
47.766667         16.766667          117.0
```

Example 3: Select station code from the station 'Birkenes II':

```
sqlite> select station_code from station where station_name='Birkenes II';
station_code
------------
NO0002R
```

Example 4: Select all files with data from station code NO0002R:

```
sqlite> select distinct filename from variable where station_code='NO0002R';
filename
-----------------------------------------------------------------------------------------
NO0002R.20090909000000.20110115000000.kfg...17w.1w.NO01L_kfg_no02.NO01L_Thermo_Optical-Sunset_Lab.lev2.nas
NO0002R.20091230070000.20110523000000.kfg.organic_carbon.pm10.53w.1w.NO01L_kfg_no02.NO01L_Thermo_Optical-Sunset_Lab.lev2.nas
NO0002R.20091230070000.20120314000000.kfg.organic_carbon.pm25.53w.1w.NO01L_kfg_no01.NO01L_Thermo_Optical-Sunset_Lab.lev2.nas
NO0002R.20091230070000.20120413000000.kfg.organic_carbon.pm25.52w.1w.NO01L_kfg_no01.NO01L_Thermo_Optical-Sunset_Lab.lev2.nas
NO0002R.20100101000000.20130101000000.uv_abs.ozone.air.1y.1h.NO01L_uv_abs_uk_0002.NO01L_uv_abs.lev2.nas
NO0002R.20100101070000.20120101000000.filter_3pack..aerosol.1y.1d.NO01L_f3p_d_0002.NO01L_IC_aero.lev2.nas
NO0002R.20100106000000.20110115000000.kfg...52w.1w.NO01L_kfg_no02.NO01L_Thermo_Optical-Sunset_Lab.lev2.nas
```

Example 5: Select all files with data from station “Birkenes II” (This combines example 3 and 4 into one statement):

```
sqlite> select distinct filename from variable
   ...>  join station on station.station_code=variable.station_code
   ...> where station_name='Birkenes II';
filename
------------------------------------------------------------------------------------
NO0002R.20090909000000.20110115000000.kfg...17w.1w.NO01L_kfg_no02.NO01L_Thermo_Optical-Sunset_Lab.lev2.nas
NO0002R.20091230070000.20110523000000.kfg.organic_carbon.pm10.53w.1w.NO01L_kfg_no02.NO01L_Thermo_Optical-Sunset_Lab.lev2.nas
NO0002R.20091230070000.20120314000000.kfg.organic_carbon.pm25.53w.1w.NO01L_kfg_no01.NO01L_Thermo_Optical-Sunset_Lab.lev2.nas
NO0002R.20091230070000.20120413000000.kfg.organic_carbon.pm25.52w.1w.NO01L_kfg_no01.NO01L_Thermo_Optical-Sunset_Lab.lev2.nas
NO0002R.20100101000000.20130101000000.uv_abs.ozone.air.1y.1h.NO01L_uv_abs_uk_0002.NO01L_uv_abs.lev2.nas
NO0002R.20100101070000.20120101000000.filter_3pack..aerosol.1y.1d.NO01L_f3p_d_0002.NO01L_IC_aero.lev2.nas
NO0002R.20100106000000.20110115000000.kfg...52w.1w.NO01L_kfg_no02.NO01L_Thermo_Optical-Sunset_Lab.lev2.nas
```

Example 6: Find all files which contain component name “arsenic” with statistic code “arithmetic mean” in any aerosol matrix, and require data in February 2008:

```
sqlite> select distinct filename from variable
   ...>  join station on station.station_code=variable.station_code
   ...> where comp_name='arsenic' and statistics='arithmetic mean' and
   ...>  matrix in ('pm1', 'pm25', 'pm10', 'aerosol') and
   ...> first_end < '2008-03-01' and last_start > '2008-02-01';
filename
------------------------------------------------------------------------------------
DK0010G.20080107000000.20101130000000.filter_3pack..aerosol.52w.1w.DK01L_f3p_10.DK01L_PIXE.lev2.nas
IS0091R.20080101110224.20101206000000.high_vol_sampler..aerosol.1y.2w.NO01L_flos1.IS02L_icpms_nilu.lev2.nas
LV0016R.20080107000000.20100702000000.b-attenuation..pm10.52w.1w.LV01L_opsis_sm200_16.LV01L_ICP_MSf.lev2.nas
DE0002R.20080101230000.20120215000000.low_vol_sampler..pm10.1y.1w.DE03L_lvs_hm_1w_02.DE03L_ICP_MS.lev2.nas
DE0008R.20080101230000.20120215000000.low_vol_sampler..pm10.1y.1w.DE03L_lvs_hm_1w_08.DE03L_ICP_MS.lev2.nas
DE0003R.20080101230000.20120215000000.low_vol_sampler..pm10.1y.1w.DE03L_lvs_hm_1w_03.DE03L_ICP_MS.lev2.nas
ES1778R.20080104120000.20100325000000.high_vol_sampler..pm25.1y.1d.ES05L_ES1778R_HVS_DIG_PM25_02.ES05L_ICPMS.lev2.nas
SK0007R.20080102000000.20120925000000.low_vol_sampler..pm10.1y.1w.SK01L_lowvol_07.SK01L_ICP_MS.lev2.nas
DE0007R.20080101230000.20090723000000.low_vol_sampler..pm10.1y.1w.DE03L_lvs_hm_1w_07.DE03L_ICP_MS.lev2.nas
FI0037R.20080107060000.20100616000000.low_vol_sampler..aerosol.52w.1w.FI01L_lws_hm37.FI01L_ICP_MS.lev2.nas
FI0017R.20080102060000.20091028000000.low_vol_sampler..aerosol.1y.2d.FI01L_lws_hm17.FI01L_ICP_MS.lev2.nas
DK0005R.20080101070000.20031001000000.filter_3pack..aerosol.1y.1d.DK01L_FP4N.DK01L_PIXE.lev2.nas
DE0009R.20080101230000.20110929000000.low_vol_sampler..pm10.53w.1w.DE03L_lvs_hm_1w_09.DE03L_ICP_MS.lev2.nas
```

Example 7: Same as in example 6, but additionally restrict the results to stations with latitude between 50 and 70 north and latidude between 0 and 20 east:

```
sqlite> select distinct filename from variable
   ...>  join station on station.station_code=variable.station_code and
   ...>   station_latitude between 50.0 and 70.0 and
   ...>   station_longitude between 0.0 and 20.0
   ...> where comp_name='arsenic' and statistics='arithmetic mean' and
   ...>  matrix in ('pm1', 'pm25', 'pm10', 'aerosol') and
   ...>  first_end < '2008-03-01' and last_start > '2008-02-01';
filename
------------------------------------------------------------------------------------
DE0002R.20080101230000.20120215000000.low_vol_sampler..pm10.1y.1w.DE03L_lvs_hm_1w_02.DE03L_ICP_MS.lev2.nas
DE0008R.20080101230000.20120215000000.low_vol_sampler..pm10.1y.1w.DE03L_lvs_hm_1w_08.DE03L_ICP_MS.lev2.nas
DE0007R.20080101230000.20090723000000.low_vol_sampler..pm10.1y.1w.DE03L_lvs_hm_1w_07.DE03L_ICP_MS.lev2.nas
DK0005R.20080101070000.20031001000000.filter_3pack..aerosol.1y.1d.DK01L_FP4N.DK01L_PIXE.lev2.nas
DE0009R.20080101230000.20110929000000.low_vol_sampler..pm10.53w.1w.DE03L_lvs_hm_1w_09.DE03L_ICP_MS.lev2.nas
```

Help and Feedback

If you experience problems, need help or want to give us feedback, please contact the EBAS support team at ebas@nilu.no.