**Creating Key Performance Indicators for the marine ICOS data**

The Ocean Thematic Centre (OTC) wish to create some Key Performance Indicators (KPI) for the marine ICOS data. These will give some insight into how the stations, sensors and instruments perform over time. The KPIs will basically be various plots which we imagine will show things like data frequency, how often a sensor fails, and how QC is performed etc. over various time scales. The exact list of plots have not yet been decided.

We will setup a system so that various reports based on the KPIs plots can be made on the fly (possibly with an easy to use GUI). The reports we have in mind includes (but are not limited to):

* *station reports*: giving insight into how a station has performed since entering ICOS. These are of interest to the Station PIs
* *annual reports*: showing how the marine stations have perfomed in the last year. OTC can include these in their annual reports to stakeholders such as NFR and the Head Office.
* *sensor reports*: combining statistics on a specific sensor type across the stations. These are relevant for the manufacturers.
* *country reports*: how does the stations from a specific counry perform. Of interest to the countries focal point and their funders.
* *QuinCe report*: can show how people QC their data, and shed light on weaknesses in the QC system, and possibly detect if QC is not performed in a similar way. This is of interest to the OTC and the further software development.

The tasks involved in creating the KPIs can be broken down to:

1. Create a KPI Database (KPIDB) containing all information we will need to create the KPIs
2. Create the KPI plots
3. Create the reports
4. GUI

The rest of this document will describe these three steps and their sub-steps in detail.

1. **Create the KPI Database (KPIDB)**

The KPIDB will be a synchronized simplified database containing the information needed to create the KPIs. It will rely on metadata from the 'OTC metadata entry' at the Carbon Portal, and QC statistics from QuinCe.

* 1. **Structure of the KPIDB**

KPIDB is a relational database created in Sqlite. See a sketch of the tables, their content and their links in the powerpoint file ‘KPIDB\_sketch\_v1.pptx’.

This database structure was set up in an sqlite database. See the SQL script ‘KPIDB\_create\_emptyTables’ for how this was created. An empty KPIDB are also stored, named: ‘KPIDB\_emptyTables.db’.

* 1. **Extracting metadata updates from the Carbon Portal**

The KPIs will somehow need to have access to the marine metadata which we currently are keeping up to date in the ['ICOS OTC metadata entry'](https://meta.icos-cp.eu/edit/otcentry/), a relational database at the CP servers.To avoid duplicating our efforts, and risk conflicting metadata, we wish to extract these metadata from the OTC metadata entry.

A python script will daily requests for updates in the OTC metadata entry since our last request. We will only need the updates since we plan to store this information in a more simplified metadata database for the purpose of the KPIs (more on this later). We wish that the output from such a request is a csv file with header:

type,id,field,value,link\_type,link\_id

type - which type has been changed  
id - what is the id of the entry that changed (the last bit of the URL)  
field - which field have change  
value - what is the new value  
link\_type - if this value represents a forreign key, which type does it belong to  
link\_id - what is the forreign key id (again, last bit of the URL)

Some exceptions:

* If links points towards types not shown in the 'Types' box at the 'ICOS OTC metadata entry' (such as 'geographical region' and 'role kind') we will not need the 'link\_type' and 'link\_id' in the csv file.
* If a new entry is created, and no fields are filled in, it is sufficient to simply give the type and id, and leave the rest as NA.

Here is an example: If we do the following changes in the OTC metadata entry:

* create a new person entry and specify their first and last name
* create a new station entry (without any more specifications)
* assign the new person as PI of the new station

... the csv file would look like this:

type,id,field,value,link\_type,link\_id  
Person,IngunnSkjelvan,first name,Ingunn,NA,NA  
Person,IngunnSkjelvan,last name,Skjelvan,NA,NA  
Station,RV\_G.O.Sars,NA,NA,NA,NA   
Assumed Role,skjelvan\_gosars,role holder,Ingunn Skjelvan,Person,IngunnSkjelvan  
Assumed Role,skjelvan\_gosars,role kind,Principal investigator,NA,NA  
Assumed Role,skjelvan\_gosars,role's organization,R/V G.O.Sars,Station,RV\_G.O.Sars

* 1. **Importing metadata to the KPIDB**

The python script ‘import\_metadata\_to\_KPIDB.py’ will import the csv file requested from the CP, and export the metadata to the KPIDB.

In order to do this, we first need to know how the structure of the OTC metadata entry database compares to the KPIDB we will create. The structure of the ‘OTC metadata entry’ and the KPIDB is similar, but not the same. Their similarities and differences had to be mapped in detail in order to convert from one to the other. The powerpoint file ‘compare\_databases.pptx’ gives an overview of the structural differences, while the excel sheet ‘linking\_databases.xlsx’ shows the conversions from one database to the other database in detail.

*More about this …?*

* 1. **Extracting QC stats from QuinCe**

*To come…*

* 1. **Extracting QC stats from QuinCe**

*To come…*

1. **Create the KPI Plots**

*To come…*

1. **Create the KPI Reports**

*To come…*

1. **Create the GUI**

*To come…*