5.1) How to extract data

How to extract all the data for a given sensor and for a given sequence in a generic object and provides the number of data ?

dataExtraction.cpp

```
#include <iostream>
#include <sstream>
#include <vector>
#include <string>

#include "containers.h"
#include "config.h"

int main(int argc, char* argv[])
{

   setlocale (LC_ALL,"C");

   //  Usage: ./DataExtract sequencePath sensorAlias
   if(argc==3)
   {
     std::string sequencePath(argv[1]);
     std::string sensorAlias(argv[2]);

     ConfigTable ctable;
     std::stringstream msg;
     msg << CONFIGFOLDERPATH << "/config_table";
     ctable.readTable(msg.str());

     std::vector<Unit> data;

     std::string
timestamps_file_name=ctable.table[sensorAlias].folderName+ctable.table[sensorAlias].suffix;
     std::string
sensor_data_folder=sequencePath+"/"+ctable.table[sensorAlias].folderName+"/";

     getReadings(ctable.table[sensorAlias].sensorName, sensor_data_folder,
timestamps_file_name, ctable.table[sensorAlias].tfm, data, TimestampInterval());

     std::cout << "================" << std::endl;
     std::cout << "There are " << data.size() << " data for the sensor " << sensorAlias << " in
the sequence stored in " << sequencePath << std::endl;
     std::cout << "================" << std::endl;
   }

  return 0;
}
```

Explanation :

- The file config_table (in the libIPDS) contains all information about sensors and logs. It is read by the object ctable, instance of a ConfigTable

```
ConfigTable ctable;
std::stringstream msg;
msg << CONFIGFOLDERPATH << "/config_table";
ctable.readTable(msg.str());
```

- The path to the file containing the timestamp for the give sensor and the path to the folder where the data of the given sensor for the given sequence are generated using information contained in ctable.

```
std::string
timestamps_file_name=ctable.table[sensorAlias].folderName+ctable.table[sensorAlias].suffix;
std::string sensor_data_folder=sequencePath+"/"+ctable.table[sensorAlias].folderName+"/";
```

- All data will be stored as a Unit object (it is a generic container).

```
std::vector<Unit> data;
```

- The data are read :

```
getReadings(ctable.table[sensorAlias].sensorName, sensor_data_folder, timestamps_file_name,
ctable.table[sensorAlias].tfm, data, TimestampInterval());
```

In that case, the time range is not specified (TimestampInterval()) thus all data will be extracted.
An other solution is to create a timestamp element and specify the starting and ending times :

```
TimestampInterval timeInt(1352474726081521,1352474722081521) ;
getReadings(ctable.table[sensorAlias].sensorName, sensor_data_folder, timestamps_file_name,
ctable.table[sensorAlias].tfm, data, timeInt);
```

The object data now contains all the Unit for the give sequence and sensor.

CMakeLists.txt

```
project(DataExtraction)
cmake_minimum_required(VERSION 2.6)

set( CMAKE_BUILD_TYPE Release )
add_definitions( -Wall )

set(libIPDS_include_dir "/home/courbon/IPDS/IPDSCodes/IPDSLib/include/")
set(libIPDS_lib_dir "/home/courbon/IPDS/IPDSCodes/IPDSLib/build/lib/")


# IPDS library inclusion
include_directories(${libIPDS_include_dir})
link_directories(${libIPDS_lib_dir})

add_executable (${PROJECT_NAME} dataExtraction.cpp)

target_link_libraries(
    ${PROJECT_NAME}
    IPDSlib
)
```

Note that the paths to the IPDS include and library (libIPDS_include_dir and libIPDS_lib_dir) have to be modified to suit your set-up.

(the source code is in the archive DataExtraction.zip)