DQA User Manual

Included files :

- ois-dqa-core-1.8.jar
- database_create.sql
- user_create.sql

Phase I : Database Setup

First you need to install the database in order to make the DQA work, the DQA is using a MySQL database, here is link if you need help to install MySQL
https://dev.mysql.com/doc/refman/5.1/en/installing.html

Once you have MySQL running, you need to run the **database_create.sql** script followed by the **user_create.sql** script

Phase II : Add Jar file

Now all you have to do is to add the Jar file to the build path of your project, all the dependencies have been compiled to this jar file.

Phase III : Use the DQA

You have to look for two classes **ProcessMessageHL7** and **CompactReportModel**
Here is an example for how to use it :

```
public static void main(String[] args) {
    try {
        /**
         * test is a string containing the HL7 message content
         */
        String test = new String(readAllBytes(get("msgtst")));

        /**
         * process is a static method of ProcessMessageHL7 it takes as parameters
         * The content of the message and a Facility id
         * For the current version the facility id will always be the string 1223
         *
         * returns a CompactReportModel
         */
        CompactReportModel crm = ProcessMessageHL7.process(test, "1223");

        System.out.println(crm);
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

The return type of the ***process*** method is a **CompactReportModel i**ts main structure consists of an ArrayList of **CompactReportNode,** each node represents an issue. You can access the ArrayList using the methode :

public ArrayList<**CompactReportNode**> getIssuesList()

The CompactReportNode structure is as follows :

```
public class CompactReportNode {

    /**
     * Issue's name containing a small description
     */
    public String issue_small_desc;
    /**
     * The location where the issue was reported
     */
    public String location;
    /**
     * Issue's type, can be Error or Warning
     */
    public String type;
    /**
     * A detailed description of the issue
     */
    public String detailed_desc;
    /**
     * The issue id
     */
    public String potential_issue_id;
    /**
     * The given code in the message if the issue concerns an invalid code
     */
    public String given_code = "N/A";
```