

Test suite (QCS 2.x)

Overview

The QCS application is provided with an embedded test suite. The test suite is composed by three different assets:

1. **Input:** The set of input test files, that is: some dummy Incidence dataset, Mortality dataset etc.
2. **Output:** The set of expected output files, that is: the reports that should be produced when running QCS on the above input files
3. **Script:** The bash script running the QCS application on all input files, and comparing each result with the corresponding expected output file

Remark: all input files contain *fake data*, that is: data created manually (not from a data submission).

Input files

The set of test input files is located in the directory:

Location of input files
<code>src\test\resources\test-data\input\encr\2022</code>

These files are divided into six different folders:

1. **incidence:** contains 5 dummy input datasets with the ENCR *Incidence 2022* data structure
2. **lifetable:** contains 2 dummy input datasets with the ENCR *Life Table 2022* data structure
3. **mortality:** contains 2 dummy input datasets with the ENCR *Mortality 2022* data structure where the **Age** variable is provided as a *value*
4. **mortality_range:** contains 2 dummy input datasets with the ENCR *Mortality 2022* data structure where the **Age** variable is provided as a *range*
5. **population:** contains 2 dummy input datasets with the ENCR *Population 2022* data structure where the **Age** variable is provided as a *value*
6. **population_range:** contains 2 dummy input datasets with the ENCR *Population 2022* data structure where the **Age** variable is provided as a *range*

These files can be used also to test the QCS application directly from the GUI: in such case - if the user wants to verify the result of the validation - she must compare the output reports produced by the QCS (in the normal *Output* folder) with the expected output reports of the test suite (see "Output files" below).

Each input file is available in two formats:

- **dataset_number.csv:** the actual test file used when running the test suite
- **dataset_number.xls:** the file used to define all test cases, and then export them in CSV file (optional)

The Excel version of each dataset is not necessary to run the test suite. These Excel files are used just to define test cases in a human-friendly way:

- **Zebra crossing layout:** different background colors are alternated to make easier to spot the set of test cases referring the same rule/feature
- **Yellow columns:** a yellow background color is often used to highlight the columns being the main focus of a test case
- **Red values:** cells colored in **red** are the wrong/missing/uncoherent values that should trigger a warning/error (or just to test a specific feature)

Usually these files are updated using the *master slave* approach:

1. First we add/edit necessary test cases in the Excel file, and we color accordingly the lines/cells being the main focus of the test case
2. Then we export the file in CSV file: this is the actual file used when running the test suite

Exception

Occasionally, some test cases could be added on the fly directly on the CSV file, without inserting them first in the Excel file. If this happens, the CSV file should have *more* rows than the corresponding Excel file. In this scenario, the *master slave* approach is reverted: the extra lines (present only in the CSV file) must be added in the Excel file, in order to synchronize the CSV file with the Excel file.

Output files

The set of expected output files is located in the directory:

Location of output files
<code>src\test\resources\test-data\output\encr\2022</code>

These files are divided into six different folders, using the same organization of the input files (see above).

Example:

Running QCS on the dataset *input/encr/2022/incidence/incidence_1.csv* should produce these reports:

- *output/encr/2022/incidence/incidence_1.csv*
- *output/encr/2022/incidence/incidence_1.txt*

Exception

The output folder contains other two directories:

- **incidence_mpm**: expected results of the input *Incidence* files when the experimental "MPMT" algorithm is enabled
- **incidence_pm_strict**: expected results of the input *Incidence* files when the "strict selection" criterion is applied to select Multiple Primaries candidates

These directories are meant to be used only for advanced testing by the JRC development team: purpose of these files will be clarified in the future.

Configuring the test suite

Before executing the test suite, some local configuration is needed.

1. Make a copy of the *test-suite-template.sh* and rename it as *test-suite.sh*
2. Open the *test-suite.sh* file with a text editor and edit the lines below

User profile in the test suite

```
# 1. SET HERE THE ABSOLUTE PATH OF THE TEST FILES
# Write your path here
DATA_HOME="C:/Home/Workspace/qcs/src/test/resources/test-data"

# 2. SET HERE THE NAME OF THE IDE PROJECT
# Write name of root directory here
PROJECT_NAME="qcs"
```

Where DATA_HOME must refer to the absolute path hosting the *input* and *output* folders of the test suite, and PROJECT_NAME must match the name of root directory hosting the QCS source code.

By default the test suite executes a test run over all supported protocols: *Incidence*, *Mortality*, *Life Table* and *Population*. This process can take from few seconds up to 2-3 minutes, depending by the power of the local machine. If necessary, it's possible to test only a single protocol or just a subset of protocols, by commenting accordingly the protocol's variables defined in the script.

Example: the configuration listed below will test only the *ENCR Incidence 2022* protocol, and will ignore all others ENCR protocols.

Selection of protocols to be tested

```
declare -a protocols=("incidence_2022")
# declare -a protocols=("incidence_2022" "incidence_2022_range" "population_2022" "population_2022_range"
"lifetable_2022")
# declare -a protocols=("incidence_2022" "population_2022" "population_2022_range" "mortality_2022"
"mortality_2022_range" "lifetable_2022")
```

Versioning

The *test-suite.sh* file is meant to be a local file, since may depend on the local machine and on the user's profile. This file must **never** be pushed on the remote GIT repository.

When the logic of the script needs to be changed or updated, the developer should first play with her own version of the local *test-suite.sh* file. Then, when the modifications are done, all changes should be merged into the *test-suite-template.sh* file, which should be pushed on the remote repository. Therefore, when a user performs a *git pull* command and notice some changes in the *test-suite-template.sh* file, she should merge the changes from the template into her local *test-suite.sh* file before running the test suite.

Running the test suite

To execute the test suite simply move into the *src/bin* directory and execute following command:

Execution of the test suite

```
./test-suite.sh
```

If everything is correct, after few minutes the result should be something like:

```
Expected reports base directory : C:/Home/Workspace/qcs/src/test/resources/test-data/output/encr

Comparing outputs for ENCR protocol "Incidence protocol (39 var)" [ID:11]
Tolerance value for comparison: 10

    2022/incidence/incidence_1 --> differences in TXT report: 0
    2022/incidence/incidence_1 --> differences in CSV report: 0
    2022/incidence/incidence_2 --> differences in TXT report: 0
    2022/incidence/incidence_2 --> differences in CSV report: 0
    2022/incidence/incidence_3 --> differences in TXT report: 0
    2022/incidence/incidence_3 --> differences in CSV report: 0
    2022/incidence/incidence_4 --> differences in TXT report: 0
    2022/incidence/incidence_4 --> differences in CSV report: 0
    2022/incidence/incidence_5 --> differences in TXT report: 0
    2022/incidence/incidence_5 --> differences in CSV report: 0

Comparing outputs for ENCR protocol "Incidence protocol (39 var)" [ID:11]
Tolerance value for comparison: 10

    2022/incidence/incidence_1 --> differences in TXT report: 0
    2022/incidence/incidence_1 --> differences in CSV report: 0
    2022/incidence/incidence_2 --> differences in TXT report: 0
    2022/incidence/incidence_2 --> differences in CSV report: 0
    2022/incidence/incidence_3 --> differences in TXT report: 0
    2022/incidence/incidence_3 --> differences in CSV report: 0
    2022/incidence/incidence_4 --> differences in TXT report: 0
    2022/incidence/incidence_4 --> differences in CSV report: 0
    2022/incidence/incidence_5 --> differences in TXT report: 0
    2022/incidence/incidence_5 --> differences in CSV report: 0

Comparing outputs for ENCR protocol "Population protocol "Age Unit" (6 var)" [ID:15]
Tolerance value for comparison: 10

    2022/population/population_1 --> differences in TXT report: 0
    2022/population/population_1 --> differences in CSV report: 0
    2022/population/population_2 --> differences in TXT report: 0
    2022/population/population_2 --> differences in CSV report: 0
```

Example of valid output of the test suite (truncated)

The history of **all runs** executed by the test suite is available in the *test-outcome.txt* file, located in the project's root.

When the QCS is modified, for example by adding a new validation rule to a protocol, updating the logic of an existing rule or just changing the layout of some output reports, then some "errors" in the test-suite are expected. In this case, the developer **MUST** check one by one all differences produced by the comparison. There should two possible scenarios:

1. If the differences are exactly as expected, the latest results of the test suite (i.e. the new output files) are copied in the *output/encr/2022/<protocol>* directory and pushed on the repository
2. If the differences are not as expected, the developer must go back to the coding task, and debug the application to fix the issue

Example:

Let's assume that the W-TOST rule is added to the Incidence protocol. When running the test suite, the result should inform the user of some errors:

```

Comparing outputs for ENCR protocol "Incidence protocol (39 var)" [ID:11]
Tolerance value for comparison: 10

2022/incidence/incidence_1 --> differences in TXT report: 26
2022/incidence/incidence_1 --> differences in CSV report: 252
2022/incidence/incidence_2 --> differences in TXT report: 0
2022/incidence/incidence_2 --> differences in CSV report: 0
2022/incidence/incidence_3 --> differences in TXT report: 0
2022/incidence/incidence_3 --> differences in CSV report: 0
2022/incidence/incidence_4 --> differences in TXT report: 0
2022/incidence/incidence_4 --> differences in CSV report: 0
2022/incidence/incidence_5 --> differences in TXT report: 0
2022/incidence/incidence_5 --> differences in CSV report: 0

Test result:

Total errors: 278

TEST FAILED

Ended QCS Test Suite : Fri Mar 6 11:59:35 2026
Total testing time : 29 seconds

```

Example of "wrong" output from the test suite

In this case the differences regards only the *incidence_1* test file (located in the input directory). Therefore the developer must compare the two files:

1. The current **expected report**, located in the *src/test/resources/test-data/output/encr/2022/incidence* directory
2. The corresponding report produced by the test suite, located in the *test/reports/2022/Incidence* directory

Comparing the two files should provide something like:

86	W-SUDA: Survival time not consistent with Date of Incidence and Date of Followup	86	W-SUDA: Survival time not consistent with Date of Incidence and Date of Followup
87	W-SUMU: Surv time+Followup Date/Month of Incidence not valid (Tab. 13)	87	W-SUMU: Surv time+Followup Date/Month of Incidence not valid (Tab. 13)
88	W-SUPt: Surgery+BoD+Autopsy+PT not valid (p. 35)	88	W-SUPt: Surgery+BoD+Autopsy+PT not valid (p. 35)
89	W-TBGR: Topography + behaviour + grade not valid (tab.5)	89	W-TBGR: Topography + behaviour + grade not valid (tab.5)
90	W-TNME: Valid TNM variables but TNM edition is not available	90	W-TNME: Valid TNM variables but TNM edition is not available
91	W-TNMM: Morphology not addressed by the Topography table used by the target T	91	W-TNMM: Morphology not addressed by the Topography table used by the target T
92	W-TNMS: Topography + TNM edition + T,N,M + Stage (not in accordance with the	92	W-TNMS: Topography + TNM edition + T,N,M + Stage (not in accordance with the
93		93	W-TOST: W-TOST: Staging system is not defined.
94	W-UNKN: Value set to missing/unknown	94	W-UNKN: Value set to missing/unknown
95		95	

Comparing expected output

And we should have also differences like this:

1347											1347										
1348											1348										
1349											1349										
1350											1350										
1351											1351										
1352											1352										
1353											1353										
1354											1354										
1355											1355										
1356											1356										

Comparing

In this case the developer must verify if the new lines (on the right) are correctly reporting a W-TOST warning for such record.

1. If this result is correct, the developer moves on and verify the next difference
2. If this result is not correct, the developer can stop comparison and should go back to the coding task

Important

The **latest results** of a test suite run are located here:

Location of latest reports produced by the test suite

test/reports

If the outcome of the test suite is correct, these reports can be ignored: they will be overwritten the next time the test suite is executed.

On the opposite, if the comparison produced some discrepancies, and such differences are as expected, then these files should be copied into the proper *src/test/resources/test-data/output* subfolder, and pushed on the remote repository.