# TEST 1

# Aim

Test the integrity of the **report.pl** program’s output CSV file.

# Synopsis

Convert the fixed string data records into CSV format import into excel and manually create the output records the perl script is to create. Compare this with the perl script to establish the code integrity of the main functionality.

# Method

## Excel Records Setup

1. Open ‘Input (2) (2).txt’ into a spreadsheet breaking up each line into columns according to the specification in System A File Specification (1) (3).pdf
2. Create filter headings for the top row
3. Define column names the same as the column headings
4. Add the following column heading this will become clear further on.
   1. Total\_Transaction\_Amount, Client, Product, RunningTotalClientProduct, Client, Product, FinalTotalClientProduct
5. Add this formula to the cells in the first new Total\_Transaction\_Amount Column as follows :- =QUANTITY\_LONG-QUANTITY\_SHORT
6. Add this formula in the Client column cells :-
   1. =CONCATENATE(CLIENT\_TYPE,CLIENT\_NUMBER,ACCOUNT\_NUMBER,SUBACCOUNT\_NUMBER)
7. Add this formula in the Product column cells :-
   1. =CONCATENATE(EXCHANGE\_CODE,PRODUCT\_GROUP\_CODE,SYMBOL,EXPIRATION\_DATE)
8. Sort all the data including the new columns by the same key fields specified in the csv-spec.txt file current default settings are :-
   1. CLIENT\_TYPE, CLIENT\_NUMBER, ACCOUNT\_NUMBER, SUBACCOUNT\_NUMBER, EXCHANGE\_CODE, PRODUCT\_GROUP\_CODE, SYMBOL, EXPIRATION\_DATE
9. In the RunningTotalClientProduct column create a running total of the Total\_transaction\_Amount column for each separate group of Client + Product column values.
10. Copy the final totals and Client, Product field values only into the last three columns. These final results are the results the program report.pl should output if it’s working correctly.

## Perl Output

1. In a new page import “Output.csv” after running the perl program *report.pl*

## Test

Compare the two excel pages to verify the programs integrity. The output should be the same excepting for columns Excel has automatically formatted which will be the NUMBER columns like ACCOUNT\_NUMBER. So the Client fields values in records.csv page will be shorter and appear like

CL 123421

Whereas the Output.csv Client value will be

CL 123400020001

When Excel concaternates the field values from the record values. It removes the zeroes.

# Unit Test Spreadsheet

See **report-unit-test-output-results.xls** for my test results.

# Test Result

Excel Result

|  |  |  |
| --- | --- | --- |
| Client | Product | FinalTotalClientProduct |
| CL 123421 | SGX FUNK 20100910 | -52 |
| CL 123431 | CME FUN1 20100910 | 285 |
| CL 123431 | CME FUNK. 20100910 | -215 |
| CL 432121 | SGX FUNK 20100910 | 46 |
| CL 432131 | CME FUN1 20100910 | -79 |

Perl report.pl Result

|  |  |  |
| --- | --- | --- |
| CL 123400020001 | SGX FUNK 20100910 | -52 |
| CL 123400030001 | CME FUN1 20100910 | 285 |
| CL 123400030001 | CME FUNK. 20100910 | -215 |
| CL 432100020001 | SGX FUNK 20100910 | 46 |
| CL 432100030001 | CME FUN1 20100910 | -79 |

PASS

# TEST2

## Aim

Test report.pl command line options

## Synopsis

Run the report.pl with the option –help

## Method

Run *report.pl –help*

## Expected Result

Should see the same output that occurs from running perldoc report.pm

## Test Result

PASS

# TEST3

## Aim

Test report.pl command line options

## Synopsis

Run the report.pl with the option –version

## Method

Run *report.pl –version*

## Expected Result

Should see the version output to stdout and the program quits

Log file output should report.

## Test Result

simon:Sun Sep 08 15:55:57 ~/Documents/AMRO-Technical\_Test $>report.pl -version

version 1.01 6Sep2019

simon:Sun Sep 08 15:56:06 ~/Documents/AMRO-Technical\_Test $>

simon:Sun Sep 08 15:56:06 ~/Documents/AMRO-Technical\_Test $>cat report.log

20190908-155606:-start report

simon:Sun Sep 08 15:57:03 ~/Documents/AMRO-Technical\_Test $>

PASS

## Possible Updates

Add extra log entry to report version output and exit.

## Severity

Trivial

# TEST4

## Aim

Test report.pl command line options

## Synopsis

Run the report.pl with the option –showfielddefs

## Method

Run *report.pl –showfielddefs*

## Expected Result

Should see the contents of the fields associative array showing the field names, their character length, start and end characters output to stdout.

Program exits without out putting the final CSV ouput.

Log file output should report.

## Test Result

**simon:Sun Sep 08 16:02:48 ~/Documents/AMRO-Technical\_Test $>report.pl -showfielddefs**

fieldname, len, start, finish

RECORD\_CODE,3,0,2

CLIENT\_TYPE,4,3,6

CLIENT\_NUMBER,4,7,10

ACCOUNT\_NUMBER,4,11,14

SUBACCOUNT\_NUMBER,4,15,18

OPPOSITE\_PARTY\_CODE,6,19,24

PRODUCT\_GROUP\_CODE,2,25,26

EXCHANGE\_CODE,4,27,30

SYMBOL,6,31,36

EXPIRATION\_DATE,8,37,44

CURRENCY\_CODE,3,45,47

MOVEMENT\_CODE,2,48,49

BUY\_SELL\_CODE,1,50,50

QUANTTTY\_LONG\_SIGN,1,51,51

QUANTITY\_LONG,10,52,61

QUANTITY\_SHORT\_SIGN,1,62,62

QUANTITY\_SHORT,10,63,72

EXCHIBROKER\_FEE\_I\_DEC,12,73,84

EXCHIBROKER\_FEE\_D\_C,1,85,85

EXCHBROKER\_FEE\_CUR\_CODE,3,86,88

CLEARING\_FEE\_I\_DEC,12,89,100

CLEARING\_FEE\_D\_C,1,101,101

CLEARING\_FEE\_CUR\_CODE,3,102,104

COMMISSION,12,105,116

COMMISSION\_D\_C,1,117,117

COMMISSION\_CUR\_CODE,3,118,120

TRANSACTION\_DATE,8,121,128

FUTURE\_REFERENCE,6,129,134

TICKET\_NUMBER,6,135,140

EXTERNAL\_NUMBER,6,141,146

TRANSACION\_PRICE\_I\_DEC,15,147,161

TRADER\_INITIALS,6,162,167

OPPOSITE\_TRADER\_ID,7,168,174

OPEN\_CLOSE\_CODE,1,175,175

FILLER,127,176,302

**simon:Sun Sep 08 16:07:28 ~/Documents/AMRO-Technical\_Test $>cat report.log**

20190908-160728:-start report

20190908-160728:-entering parsefieldnames

20190908-160728:-exiting parsefieldnames

20190908-160728:-entering showfielddefinitions

20190908-160728:-exiting showfielddefinitions

simon:Sun Sep 08 16:08:04 ~/Documents/AMRO-Technical\_Test $>

PASS

# TEST5

## Aim

Test report.pl command line options

## Synopsis

Run the report.pl with the option –showrecords

## Method

Run *report.pl –showrecords*

## Expected Result

Records in name equals value pairs

## Result

simon:Sun Sep 08 16:16:48 ~/Documents/AMRO-Technical\_Test $>report.pl -showrecords|more

recno 1 RECORD\_CODE=315

recno 1 CLIENT\_TYPE=CL

recno 1 CLIENT\_NUMBER=4321

recno 1 ACCOUNT\_NUMBER=0002

recno 1 SUBACCOUNT\_NUMBER=0001

recno 1 OPPOSITE\_PARTY\_CODE=SGXDC

recno 1 PRODUCT\_GROUP\_CODE=FU

recno 1 EXCHANGE\_CODE=SGX

recno 1 SYMBOL=NK

recno 1 EXPIRATION\_DATE=20100910

recno 1 CURRENCY\_CODE=JPY

recno 1 MOVEMENT\_CODE=01

recno 1 BUY\_SELL\_CODE=B

recno 1 QUANTTTY\_LONG\_SIGN=

recno 1 QUANTITY\_LONG=0000000001

recno 1 QUANTITY\_SHORT\_SIGN=

recno 1 QUANTITY\_SHORT=0000000000

recno 1 EXCHIBROKER\_FEE\_I\_DEC=000000000060

recno 1 EXCHIBROKER\_FEE\_D\_C=D

recno 1 EXCHBROKER\_FEE\_CUR\_CODE=USD

recno 1 CLEARING\_FEE\_I\_DEC=000000000030

recno 1 CLEARING\_FEE\_D\_C=D

recno 1 CLEARING\_FEE\_CUR\_CODE=USD

recno 1 COMMISSION=000000000000

recno 1 COMMISSION\_D\_C=D

recno 1 COMMISSION\_CUR\_CODE=JPY

recno 1 TRANSACTION\_DATE=20100820

recno 1 FUTURE\_REFERENCE=001238

recno 1 TICKET\_NUMBER=0

recno 1 EXTERNAL\_NUMBER=688032

recno 1 TRANSACION\_PRICE\_I\_DEC=000092500000000

recno 1 TRADER\_INITIALS=

recno 1 OPPOSITE\_TRADER\_ID=

recno 1 OPEN\_CLOSE\_CODE=O

recno 1 FILLER=

recno 2 RECORD\_CODE=315

recno 2 CLIENT\_TYPE=CL

recno 2 CLIENT\_NUMBER=4321

recno 2 ACCOUNT\_NUMBER=0002

recno 2 SUBACCOUNT\_NUMBER=0001

recno 2 OPPOSITE\_PARTY\_CODE=SGXDC

recno 2 PRODUCT\_GROUP\_CODE=FU

simon:Sun Sep 08 16:19:55 ~/Documents/AMRO-Technical\_Test $>cat report.log

20190908-161955:-start report

20190908-161955:-entering parsefieldnames

20190908-161955:-exiting parsefieldnames

20190908-161955:-entering parsedatafile

20190908-161955:-exiting parsedatafile

20190908-161955:-entering showrecords 0

20190908-161955:-exiting showrecords

## COMPARISON WITH DATA RECORD

**315**CL **4321**0002**0001**SGXDC **FU**SGX **NK 20100910**JPY**01**B **0000000001**\_**0000000000**000000000060**D**USD000000000030D**USD**000000000000**D**JPY**20100820**0012380 688032**000092500000000** **O**

## RESULT

The same.

PASS

# TEST5

## Aim

Test report.pl command line options

## Synopsis

Run the report.pl with the option –showrecordscsv

## Method

Run *report.pl –showrecordscsv > records.csv*

## Expected Result

Records in CSV format

## Result

## simon:Sun Sep 08 16:28:43 ~/Documents/AMRO-Technical\_Test $>report.pl -showrecordscsv > records.csv

**simon:Sun Sep 08 16:52:21 ~/Documents/AMRO-Technical\_Test $>cat report.log**

20190908-165221:-start report

20190908-165221:-entering parsefieldnames

20190908-165221:-exiting parsefieldnames

20190908-165221:-entering parsedatafile

20190908-165221:-exiting parsedatafile

20190908-165221:-entering showrecords csv

20190908-165221:-exiting showrecords

See records.csv compare with report-unit-test-output-results.xls Input (2) (2).txt page

PASS

# TEST6

## Aim

Test report.pl command line options

## Synopsis

Run the report.pl with the option –output2both

## Method

Run

*report.pl –output2both > test6.csv*

*diff Output.csv test6.csv*

*Should be the same*

## Expected Result

## Result

simon:Sun Sep 08 17:02:49 ~/Documents/AMRO-Technical\_Test $>report.pl -output2both > test6.csv

simon:Sun Sep 08 17:03:16 ~/Documents/AMRO-Technical\_Test $>diff Output.csv test6.csv

simon:Sun Sep 08 17:03:28 ~/Documents/AMRO-Technical\_Test $>more test6.csv

Client\_Information,Product\_Information,Total\_Transaction\_Amount

CL 123400020001,SGX FUNK 20100910,1

CL 123400020001,SGX FUNK 20100910,2

CL 123400020001,SGX FUNK 20100910,3

CL 123400020001,SGX FUNK 20100910,4

CL 123400020001,SGX FUNK 20100910,5

…

## PASS