

Test Plan

Overview..... 2

In of Scope 2

Out of Scope 2

Environment 2

Approach 2

Assumptions 2

Testing Process 3

 Unit Testing..... 3

 Functional Testing..... 4

 Performance Testing..... 5

 Integration Testing..... 5

Plan History..... 5

Overview

The purpose of the program is to simulate the way a cheque program converts a decimal number into the corresponding text.

For example:

Input: "123.45"

Output: "ONE HUNDRED AND TWENTY-THREE DOLLARS AND FORTY-FIVE CENTS"

In of Scope

- Unit Testing
- Functional Testing
- Performance Testing
- Integration Testing

Out of Scope

- Regression testing (N/A)

Environment

Attributes	Requirements
Operating system	Windows XP (32 and 64 bits) SP1 to Windows 8
.NET Framework	4.5

Approach

The purpose of this test plan is to validate the correct conversion of a decimal number into text format in terms of money accuracy.

Assumptions

- The program just accepts two decimal positions, considering that the maximum value for cents is represented in two digits.
 - 0.1 valid number in terms of money, representing ten cents
 - 0.01 valid number in terms of money, representing one cents
 - 00.001 invalid
- In terms of integers, a valid value should be a positive or negative number with 28 digits as maximum.

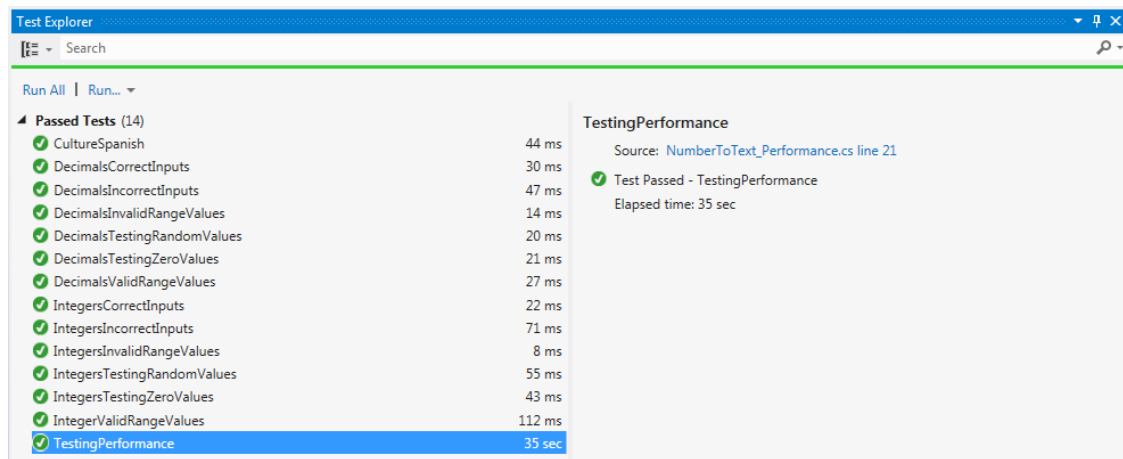
Testing Process

Unit Testing

In order to run the test methods, open the test explorer in Visual Studio and select “Run All”. The test data files that are used in the test methods could be found in the following folder “source code\ConverterUnitTesting\TestData\”, the methods cover the following scenarios:

[illegible]

CultureSpanish	SpanishData.xml	seventy five million and four hundred seventy thousand and one dollars and fifty-eight cents Input: 20450.25 Expected Result: veinte mil y cuatrocientos cincuenta dolares y veinte cinco centavos
TestingPerformance	Automatic generation of 1000 decimal numbers	Input: Expected Result: ConverterPerformanceResults.txt



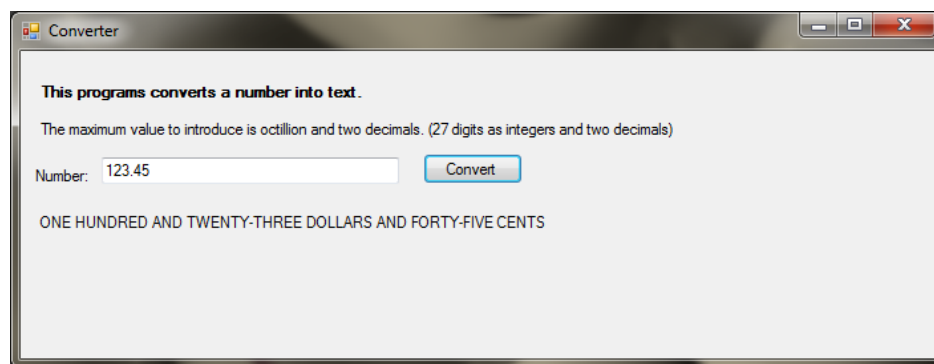
Unit testing methods

Functional Testing

Test.

The Converter desktop application was created to cover the functional testing of the cheque program. In order to run the application download the folder “application” and follow these instructions:

1. Run the executable file **ConverterUI.exe**
2. Type a decimal number on the text field.
3. Click the “Convert” button.



Converter desktop application

Note: The ConverterUI.exe needs the following components to work: ConverterResources.dll, Converter.dll

Expected Result.

English text format of the decimal number input.

Performance Testing

Test:

In order to test the performance of the converter; in the test explorer in Visual Studio run the "TestingPerformance " function that generates 10,000 random numbers within the valid ranges.

Expected result:

A file named *ConverterPerformanceResults.txt* at the same level of the **ConverterUI.exe**

Integration Testing

Test:

The converter is a class library implemented in C#; however, it could be used in a desktop application as it's currently shown in the functional testing.

Expected Result:

The user should be able to use the Converter.dll through the ConverterUI.exe interface.

Plan History

Version	Developer	Tester	Date	Comments
1.0	Brenda Jimenez		08/09/2013	Creation of the test plan.
2.0	Brenda Jimenez		10/09/2013	Updating the test plan to add the word "and" and "-" so the number written in user interface application is in the following way: 123.45 ONE HUNDRED AND TWENTY-THREE DOLLARS AND FORTY-FIVE CENTS