Project IV: Mobile & Networked Environments

Assignment #1 – Static & Dynamic Testing

Overview

In the lecture this week we reviewed Static and Dynamic testing concepts/theory. This assignment is designed to get you thinking and designing for these types of testing campaigns After all, Project IV looks at the "System Level" which means everything. 😊

You will be required to perform a code review (Static Test) and define a set of Branch and Conditional Statement coverage. Following your review, you (as a member of the software test team) will design and build a unit testing suite using MS Test (Dynamic Test) in Visual Studio.

What to do

Step #1: Clone the Workspace

There is a starting set of C++ production code for you. This production code is a simple configuration file loader. It loads a basic ASCII text file (provided within the solution space), creating data object elements, storing them in a sequential vector space and setting a set of flags and parameter values.

The first thing you need to do is clone the workspace to your own desktop using the URL provided on eConestoga.

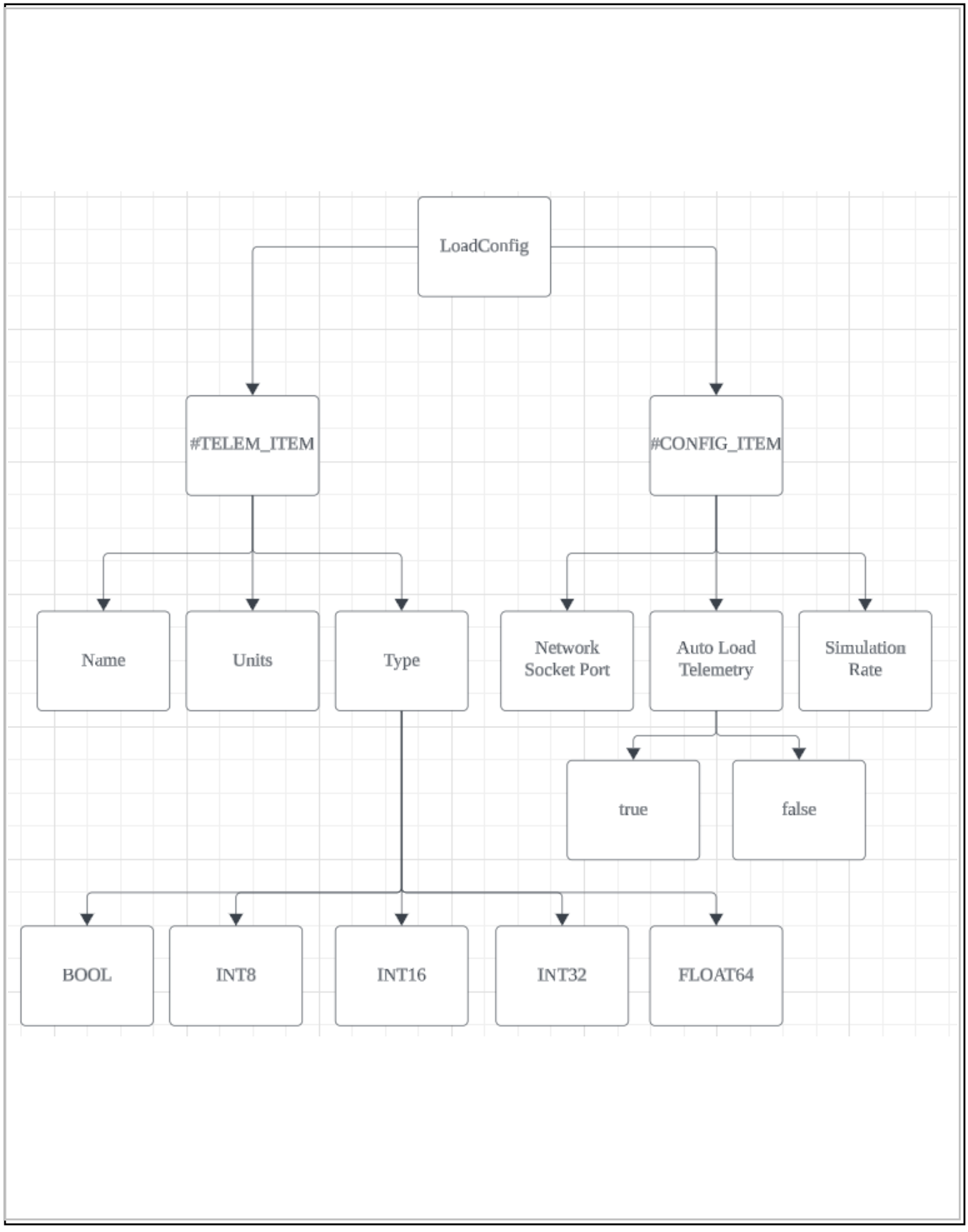
Step #2: Static Analysis

As a member of the software test development team, you have been tasked to perform a code review (Static Testing concept) for the `bool LoadConfig(std::string Filename)` function of the program.

More specifically, you will need to design and develop a test suite that will provide testing of all the branches and conditions within this function. Before you can design anything, it is important to analyze the code and determine the number of branches/conditions that need to be evaluated.

As we did for the example in class, you are to create a flow diagram showing all possible branches and conditions that need to be tested for this function. Draw your flow diagram using any tool you wish (or hand draw and scan). When finished upload an image of your flow diagram in the box below.

Static Code Review Results – Analysis of Branch/Conditions



Step #3: Dynamic Testing – Unit Test Suite

Now that you have completed the Static testing (code analysis) it's time to design and implement a set of unit tests using the MS Test Framework. Inside the solution space, create a new MS Test project and develop a complete set of unit tests to cover all the branches/conditional statements defined in your static analysis above.

NOTE: Do not ask your professor what the minimum number of unit tests are expected. You are not being evaluated based on quantity but based on quality of your test suite design. Ask yourself the following question:

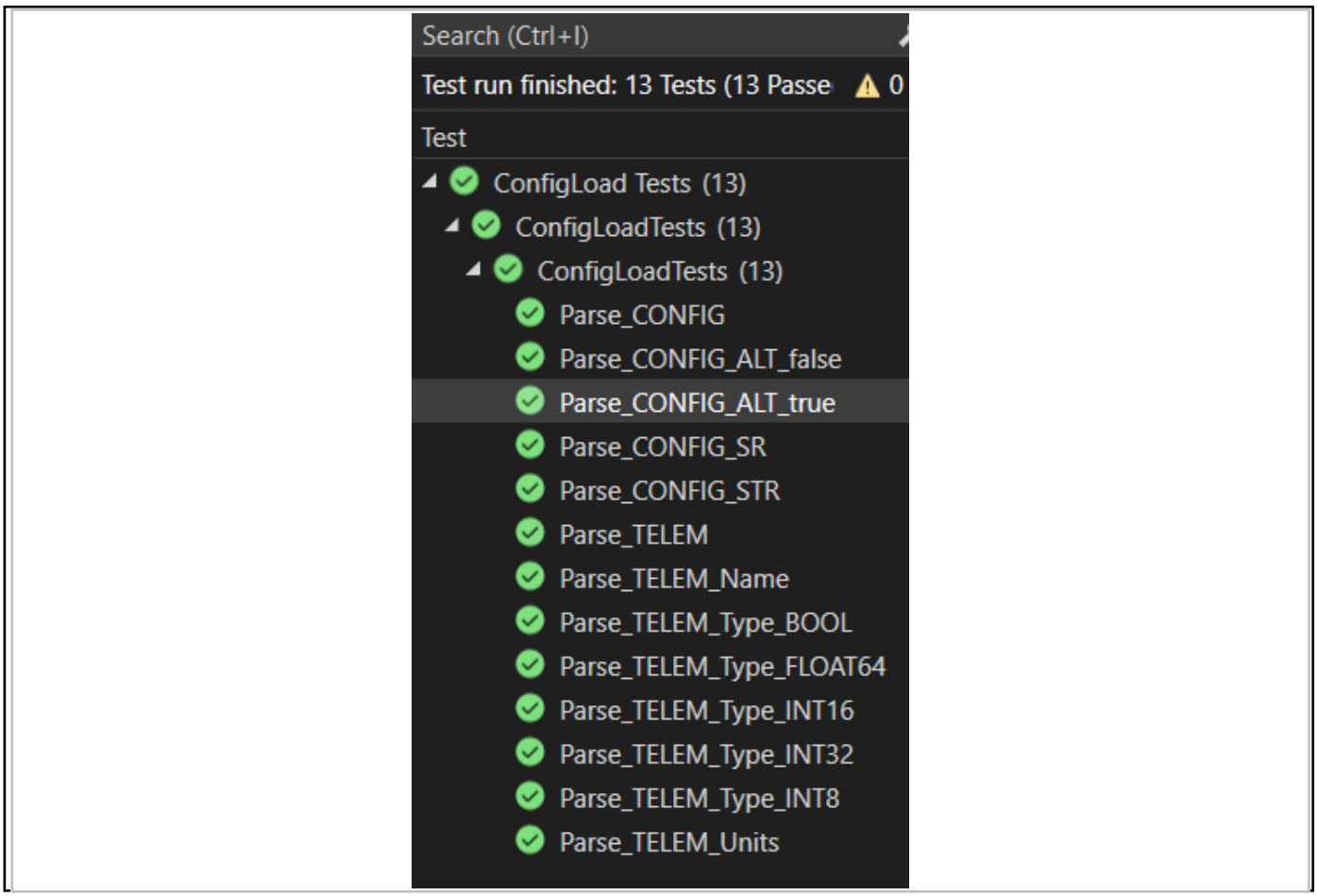
- *Does my unit testing suite cover all conditions and branches defined in the output of my static analysis?*

I recommend going back to your first semester notes to remind yourself how to externally link functions and variables between the projects.

Step #3: Dynamic Testing Results

Once you have completed the implementation of your unit testing suite, take a screenshot of your Test Explorer showing a successful pass of all your unit tests and place the image in the box below. See Appendix for example.

NOTE: Collapse all the Unit Classes in your Test Explorer window to allow for all tests to be show in a single screenshot. Remember, if one unit test fails, the unit class will show a failure as well.



Rubric

See eConestoga for details.

What to Hand In

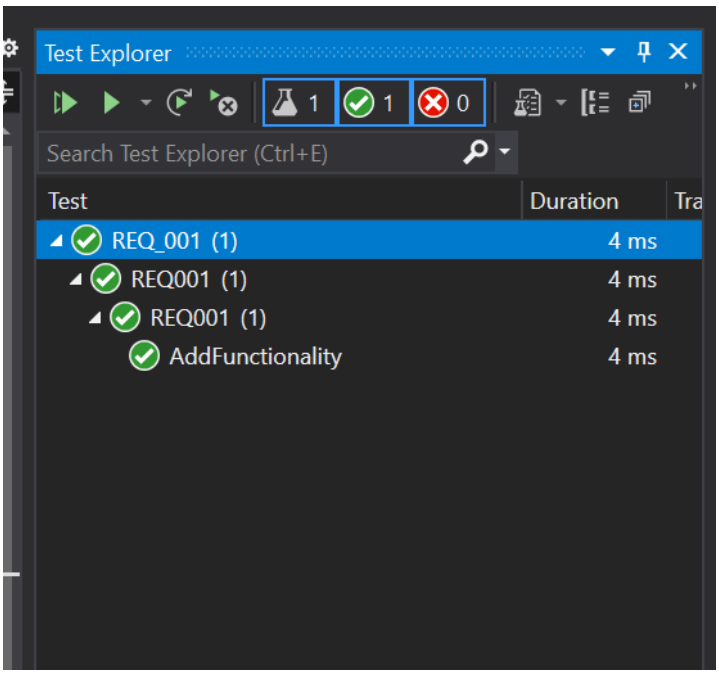
When finished, upload the following files to eConestoga:

1. A copy of this completed PDF form (unmodified).
2. A ZIP of your Visual Studio Solution Space with the Unit Testing suite completed
    a. If possible, delete all **Debug** and **Debug/x64** directories before creating the ZIP file
    b. If you have configuration data files in the debug directories, make sure you copy them into the source code directories before you delete.

KEEP A COPY!

Make sure you keep a copy of this assignment secure. You will need it as part of Assignment #2 next week.

Appendix


Test Explorer
1
1
0
Search Test Explorer (Ctrl+E)
Test
Duration
Tra
REQ_001 (1)
4 ms
REQ001 (1)
4 ms
REQ001 (1)
4 ms
AddFunctionality
4 ms