

Test Insight in Lazarus

Michaël Van Canneyt

November 20, 2023

Abstract

The FPCUnit support in Lazarus has received an upgrade: The FPCUnit unit testing framework now can communicate directly with the Lazarus IDE, making it even more easy to fix errors in your code.

1 Introduction

Free Pascal has had a unit testing framework since almost 20 years: FPCUnit is in use for the testing of many of the packages included in Free Pascal. It is roughly compatible to DUnit, the Delphi unit testframework: vanilla test code written for Dunit will compile with Free Pascal.

The Lazarus IDE has some wizards to make a FPCUnit test case and to start a FPC unit test program. 2 kinds of test program can be created: a console test program, and a GUI test program. The former shows the results of all tests on the console. The latter uses a GUI window with a treeview to show all tests and the result of the tests.

Both these programs have the same drawback: they do not interact with the IDE. It would be much nicer if you could just click on the test name and be taken to the implementation of that test in the IDE, Or, if there is a stack trace, on the error location and be taken to the location where the error is raised.

Well, now you can: In the trunk version of Lazarus, the LazFPCUnit package has been extended with 'Test Insight'. The idea for this feature has been taken from the similarly named Delphi plugin by Stefan Glienke.

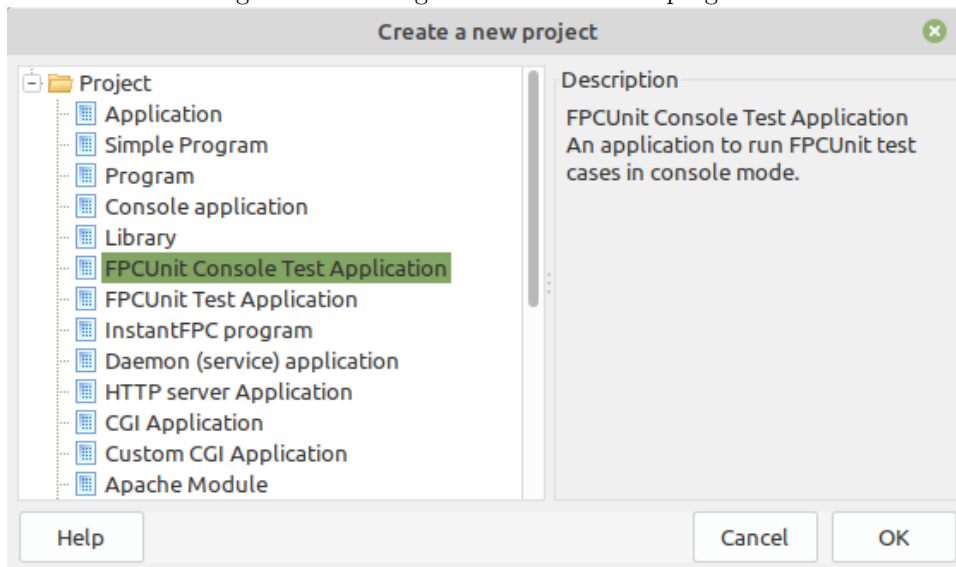
2 Architecture

The FPCUnit unit test system works like all other test systems. There is a test registry, and there are test listeners. When the tests from the registry are run, the progress and results are reported by the registered listeners. The console listeners just write to the console in one of several formats (you can register your own), while the UI listener adds nodes to a treeview.

The testinsight listener sends the results with HTTP requests to a server. The location of this server can be specified in the sources, or in a small configuration file.

The TestInsight support in the Lazarus IDE launches a small HTTP server which listens for the HTTP requests with the test results. The http server is started when you open the 'Test insight' window from the 'View - Test insight' menu in the Lazarus IDE. When the test program exists, it starts it with a special option, to

Figure 1: Creating a FPCUnit console program



get a list of tests. When you ask for a test run, the tests are run, and the window displays the test results. Double clicking on a test will use the IDE code tools to jump to the correct method in the test project.

If the test program cannot reach the testinsight HTTP server, then it will fall back to running as a regular FPCUnit test console program. You could do the same with a UI program if so desired, but this option has not been activated for the FPCUnit UI test program.

3 Usage

For this to work, you need to have the latest IDE sources, and you must of course install the lazfpunit package. it is part of the standard list of packages in the IDE. The 'New project' - 'FPCUnit Console Test Application' menu item (figure 1 on page 2) has now a project wizard which offers some options. The following options are available:

Run all tests by default The standard FPC unit console program shows a help when running it without command-line options. If this option is checked, the program will run the tests instead when no command-line options are specified.

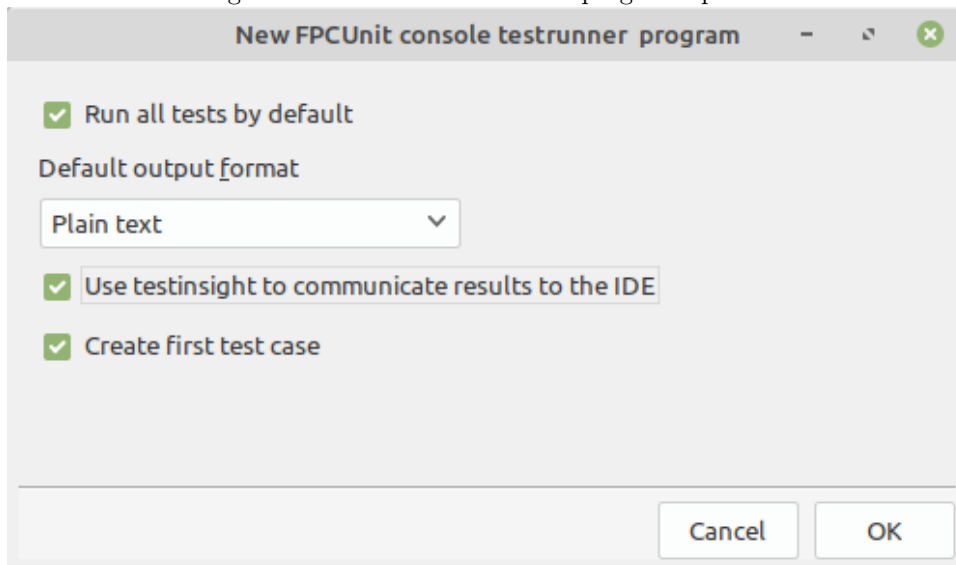
Default output format in this checkbox, you can set the default output format for the console output. You can choose between XML, plain text (with or without time info) or LaTeX.

Use testinsight to communicate results to the IDE When this option is set, the test results are sent to the IDE with testinsight.

Create first test case When checked, the IDE immediatly launches the 'Create FPCUnit testcase' wizard when the project is created.

The options dialog is shown in figure 2 on page 3. Once your project is created

Figure 2: The FPCUnit console program options



with the 'Use testinsight' option, you should save and compile it at once. This will allow the IDE testinsight support to get a list of tests.

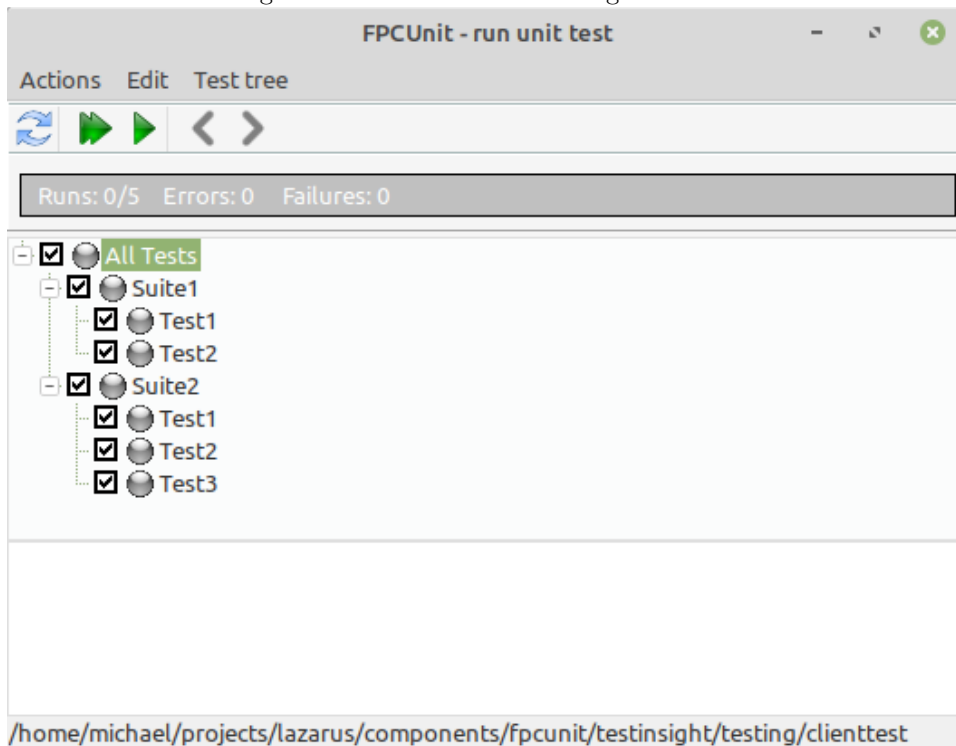
Once the project is compiled, you can open the testinsight window. The window looks the same as the window of the graphical FPCUnit test program - not surprisingly, since it was copied from that window. But some elements were added, and it behaves differently from the original window.

When the window is opened, it will attempt to run the current project with the appropriate options to get a list of test, and the result looks like figure 3 on page 4 if all went well. The 'Refresh' button (first button on the toolbar) can be used to refresh the list of tests if you want. The list of tests will be refreshed in each case when the test program is actually run. When the active project is changed, then the list of tests will also be automatically refreshed: the path of the current FPC unit test executable is always shown in the status bar at the bottom of the window.

When you click the 'Run all tests' button (the green double arrow), the test program will be executed. as the test results come in, the colored status indicator in front of each test will change color according to the result of the test. If there is an error, more info will be displayed in nodes below the test node, as shown in figure 4 on page 5. Double clicking on the test or an error node, will attempt to locate the code of the test and open the source file. For most cases, this will work without problem. However, the mechanism can fail in some cases:

1. The currently active project in the IDE is not a unit test project.
2. Item the sources of the test are not part of the project.
3. If you are using some more advanced features of the test framework (creating tests dynamically). TestInsight supposes that the test names are method names of a class.

Figure 3: The Lazarus Test Insight window



4 Converting an existing FPCUnit test program to use TestInsight

The 'New program' wizard cannot convert an existing program to use testinsight. If you have an existing FPCUnit test program that you wish to convert so it uses testinsight, you need to make some changes to the main project file.

The typical FPCUnit test console program has the following main project file source:

```
program clienttest;

{$mode objfpc}{$H+}

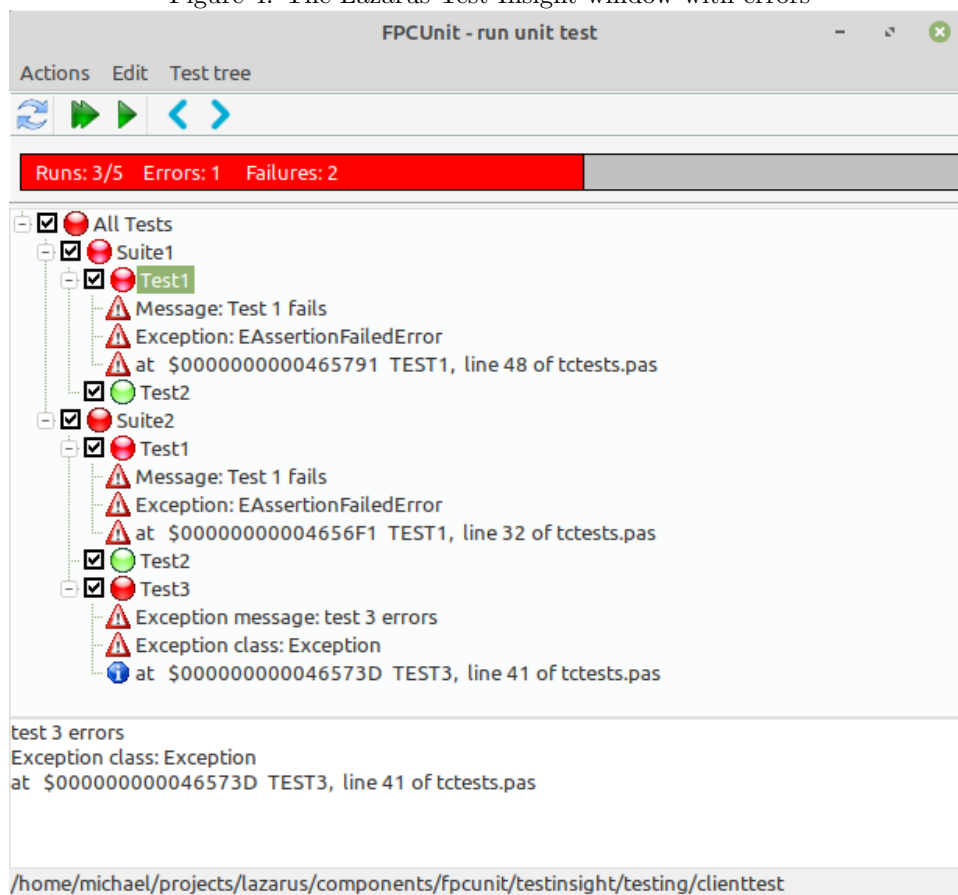
uses
    Classes, jsonparser, consoletestrunner, tcTests;

type
    TMyTestRunner = class(TTestRunner)
    end;

var
    Application: TMyTestRunner;

begin
    Application := TMyTestRunner.Create(nil);
    Application.Initialize;
    Application.Title := 'FPCUnit Console test runner';
```

Figure 4: The Lazarus Test Insight window with errors



```

    Application.Run;
    Application.Free;
end.

```

2 things must be done with this project code:

1. The `fpcunittestinsight` unit must be added to the uses clause.
2. the function `IsTestinsightListening` must be called. If it returns `True`, then the `RunRegisteredTests` routine must be called. Both are implemented in the `fpcunittestinsight` unit. If `IsTestinsightListening` returns `False`, then the original code must be executed.

The 2 functions to use are declared as follows:

```

procedure RunRegisteredTests(aConfig : String = '';
                             baseUrl: string = DefaultUrl);
function IsTestinsightListening(aConfig : String = '';
                                baseUrl: string = DefaultUrl) : Boolean;

```

The config argument is the name of an .INI file with the settings for the test run. By default this is `TestInsightSettings.ini`, this is what the IDE uses. The selected tests will be written in this file, and the port on which the IDE is listening (The `baseURL`)

The `BaseURL` argument is the URL where the testinsight server is listening. By default this is `http://localhost:8081/tests`, but this will be set by the IDE in the config file.

You could change these defaults for example to run the test program remotely but still receive the results locally on your development PC.

With these changes, the new project code will look like this:

```

program clienttest;

{$mode objfpc}{$H+}

uses
    Classes, jsonparser, consoletestrunner,
    tcTests, fpcunittestinsight;

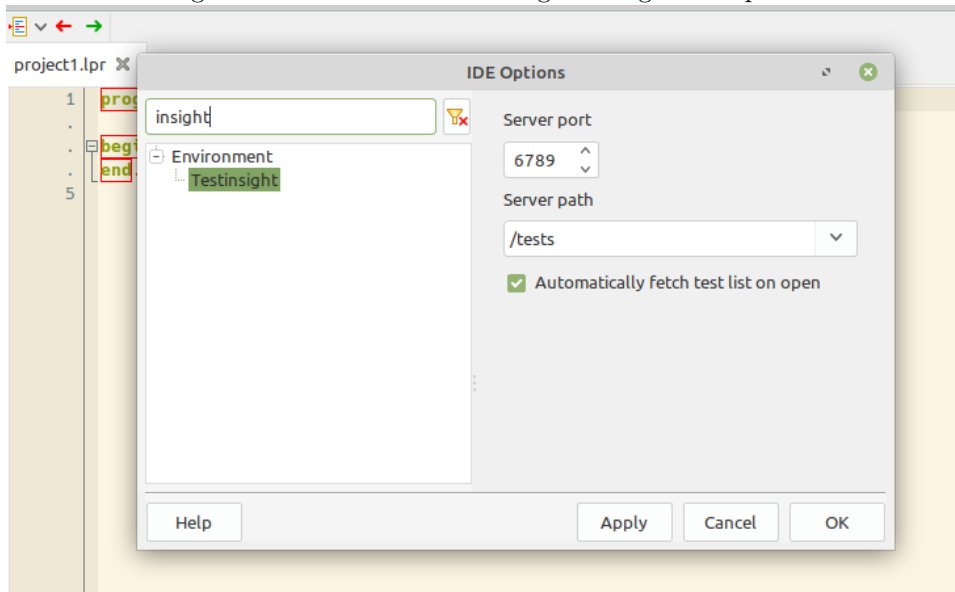
type
    TMyTestRunner = class(TTestRunner)
    end;

var
    Application: TMyTestRunner;

begin
    if IsTestinsightListening() then
        RunRegisteredTests('', '')
    else
        begin
            Application := TMyTestRunner.Create(nil);

```

Figure 5: The Lazarus Test Insight configuration panel



```
Application.Initialize;  
Application.Title := 'FPCUnit Console test runner';  
Application.Run;  
Application.Free;  
end;  
end.
```

Additionally, if you are running version 3.2.2 of the FPC compiler, in the project inspector, you should also add the 'laztestinsight' package as a requirement. For FPC 3.3.1, this dependency is not needed, as the necessary units are present in the Free Pascal sources.

5 Configuration

In the IDE, the port and base location on which the server should listen can be set. In the 'Tools - Options' menu under TestInsight, the following settings can be configured:

Base path this is the path on the HTTP server to which requests must be sent. The default is '/tests'.

Server port this is the path on the HTTP server to which requests must be sent. The default is 6789

Auto fetch tests When the 'Test Insight' window is opened, or the current project changes then the list of tests is automatically fetched. When disabled, you can use the refresh button to get the list of tests.

The settings dialog is shown in figure 5 on page 7.

6 Conclusion

The 'Test Insight' functionality makes it easier to do test-driven development by allowing you to immediatly jump to the implementation of a test or a error location from within the test result display embedded in the IDE. The current version of 'Test insight' is but an initial version: Some extensions are planned, such as test filtering and automatic selection of tests, and pas2js support of testinsight is also in the works.