Manual SpecFlow

Prerequisites

- Visual studio
- .NET framework (4.0 or higher)

Installation

You should install specflow to both the visual studio IDE(once) and every test project.

Install to IDE

1. Start Visual studio
2. Go to Tools > Extensions and updates
3. Install the Specflow nugget

Install to project

1. Add a unit test-project to your solution
2. Go to Tools > Nuget Package Manager > Manage Nuget Packages for Solution
3. Install the Specflow nugget

There are some additional tools that are not required to use specflow. But will make your output much more readable.

- SpecRunner for SpecFlow testrunner(available as nugget)

To generate output without results in various formats(usefull for disucussing the specifications).

- Pickels(http://www.picklesdoc.com/#!index.md)

Set-up

If you need to setup a language different than English to write the specifications, you will need to change this in the App.config file of your test project.

For example for Dutch:

```
<language feature="nl-NL" />
```

If you use the SpecRun testrunner, you should make sure that this is defined in the App.config of your test project.

```
<unitTestProvider name="SpecRun" />
<plugins>
      <add name="SpecRun" />
</plugins>
```

Finally delete the standard unit test from the test project.

Process

1. Add a SpecFlow feature file to the project.
2. Write the specifications for the feature.
    1. Define the feature
    2. Define the scenarios
    3. Define for each scenario the Given-When-Then's(in the defined language)

    - Feature, Scenario, ... are included in intellisense(press ctrl + space for autocomplete list).
    - It is essential you provide a title for the feature(Feature: Featuretitle). Without it, you get errors.
3. Provide tags if needed for test filtering
4. If you use pickels for documentation and discussion, follow following steps
    1. Start pickle GUI(or run tool via command line)
    2. Point out the feature file
    3. Point out the output file
    4. Selected output format
    5. Generate documentation

    - Pickels allow to add the test-results. But this generator doesn't work properly
5. Press right mouse > Generate step definitions
    - You can create a new file or copy methods to clipboard.
    - The auto-generation of the step definitions doesn't contain refactoring, meaning renames and deletes are not propagated to the step file;
    - You can add a Step definitions file manually and paste the methods from the clipboard.
    - Signals when all steps are bound
    - The generator can drop the ball when regular expressions are used, so check all the steps after generations.
    - Steps are global for the entire project. Thus group global steps in a separate file to keep the structure maintainable.
6. Check the step file for non-conforming steps and provide logical names for variables.
7. Implement the steps.
    - You can decorate your methods with hooks to allow for set-up and teardown, and other functionality
8. Run the specflow scenario's via right mouse button > run specflow scenario's. You can select which scenario to run by placing the mouse on a certain part.
    - Mouse on scenario > only that scenario.
    - Mouse on the feature > all the scenario's in the feature.
    - Mouse on the project > it will run all the features in the project.
9. If specrun is used, you can view the results in the output window.
    - Specrun introduces a variable delay(max 14s) in the free version, to indicate that the registerd version should be purchased for professional use.
    - The output window provides a link to a html page with a detailed result of the tests.

Extra info

http://www.specflow.org/getting-started/
http://www.specflow.org/documentation/
http://www.youtube.com/watch?v=SVwBz-ZTliw