# Introduction

This document describes how to use SpecFlow with Mono and MonoDevelop. We assume you are familiar with setting up a Mono environment and running MonoDevelop. You should also read the SpecFlow Guide.pdf document in conjunction with this document to retrieve any SpecFlow specific information.

# Requirements

Mono 2.6.5

Which isn't release yet, this release will include a patch that I submitted to a bug that I found.

MonoDevelop 2.4

The current release is Beta 2. The production release will be very soon.

# Installation

We are not quite ready to fill this section in. Once we have everything in place for the next release, I will fill it in. Basically, files in place for download.

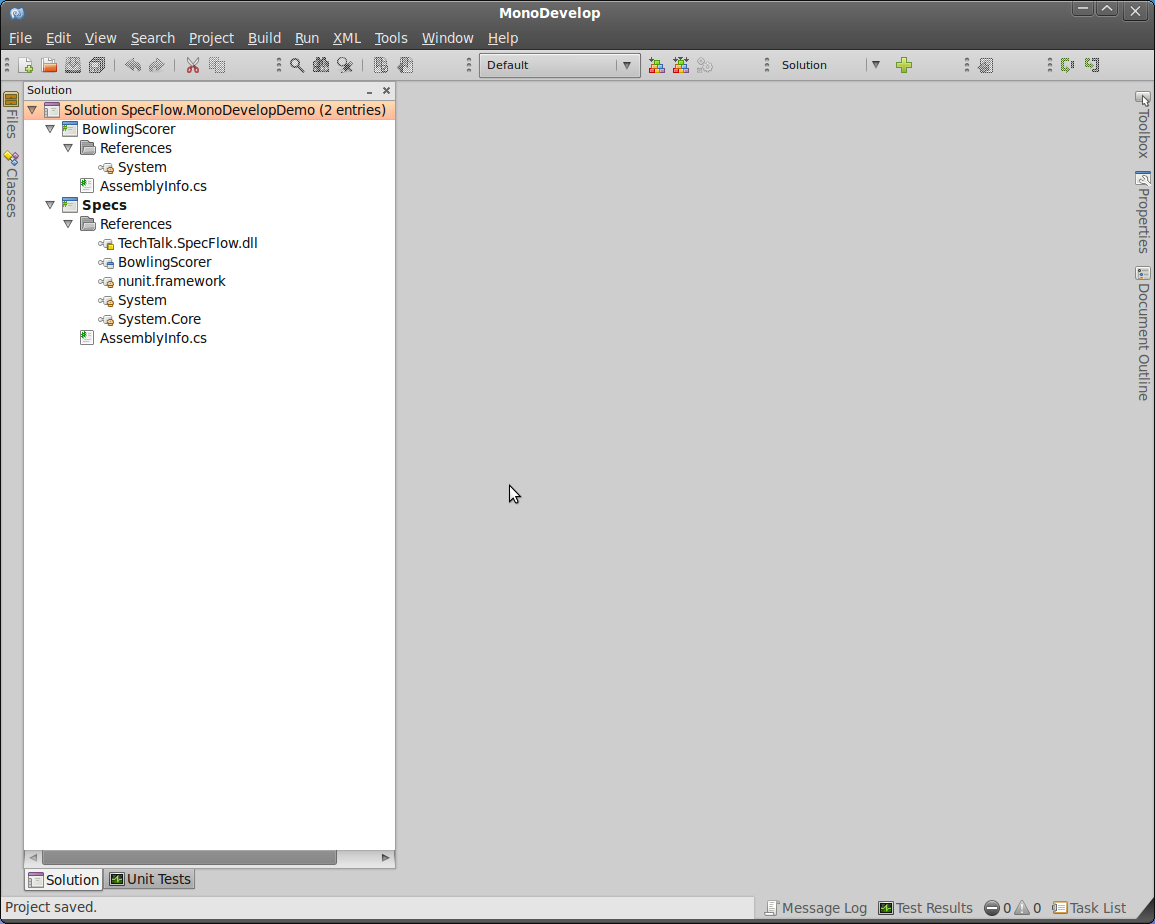
# Using the MonoDevelop Add-In

First thing that we are going to notice that there isn't a special project for working with SpecFlow. We would start developing like any other project. I am going to walk you through some steps with using SpecFlow inside of MonoDevelop and provide screenshots along the way.

1. Create a new solution called SpecFlow.MonoDevelopDemo.sln
2. Add two C# class libraries called BowlingScorer and Specs.
3. Add these references to the Specs project:

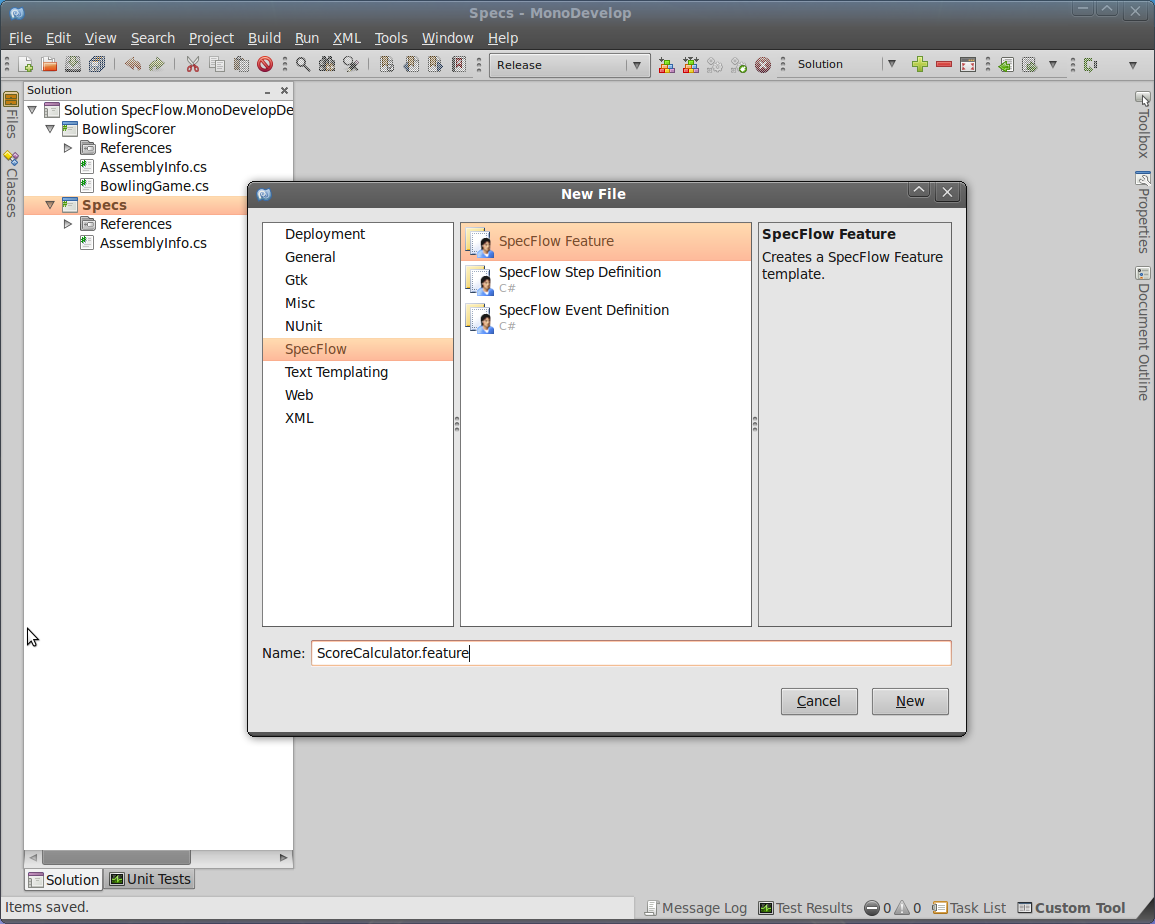
* System
* System.Core
* nunit.framework
* TechTalk.SpecFlow
* BowlingScorer

You should now have something that looks like this:

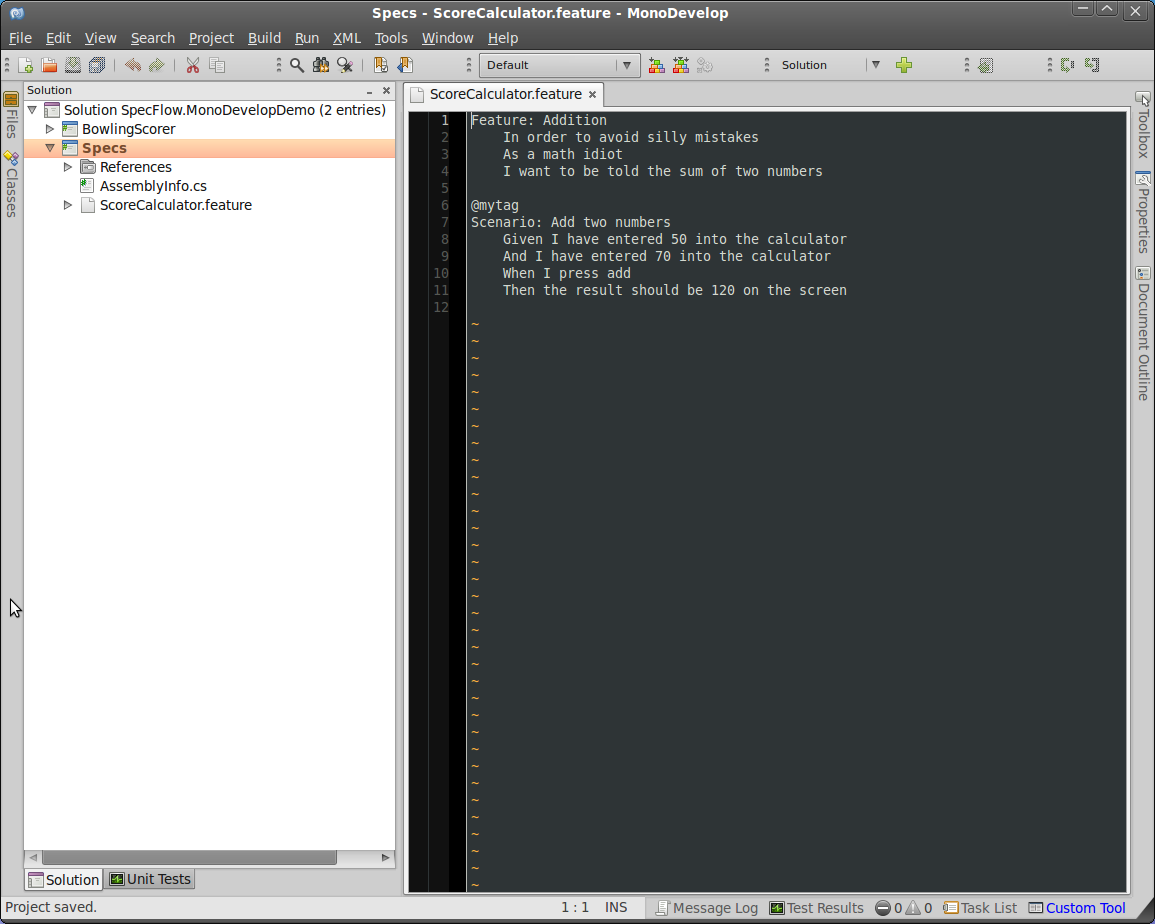


Now it's time to add a new feature to our Specs project.

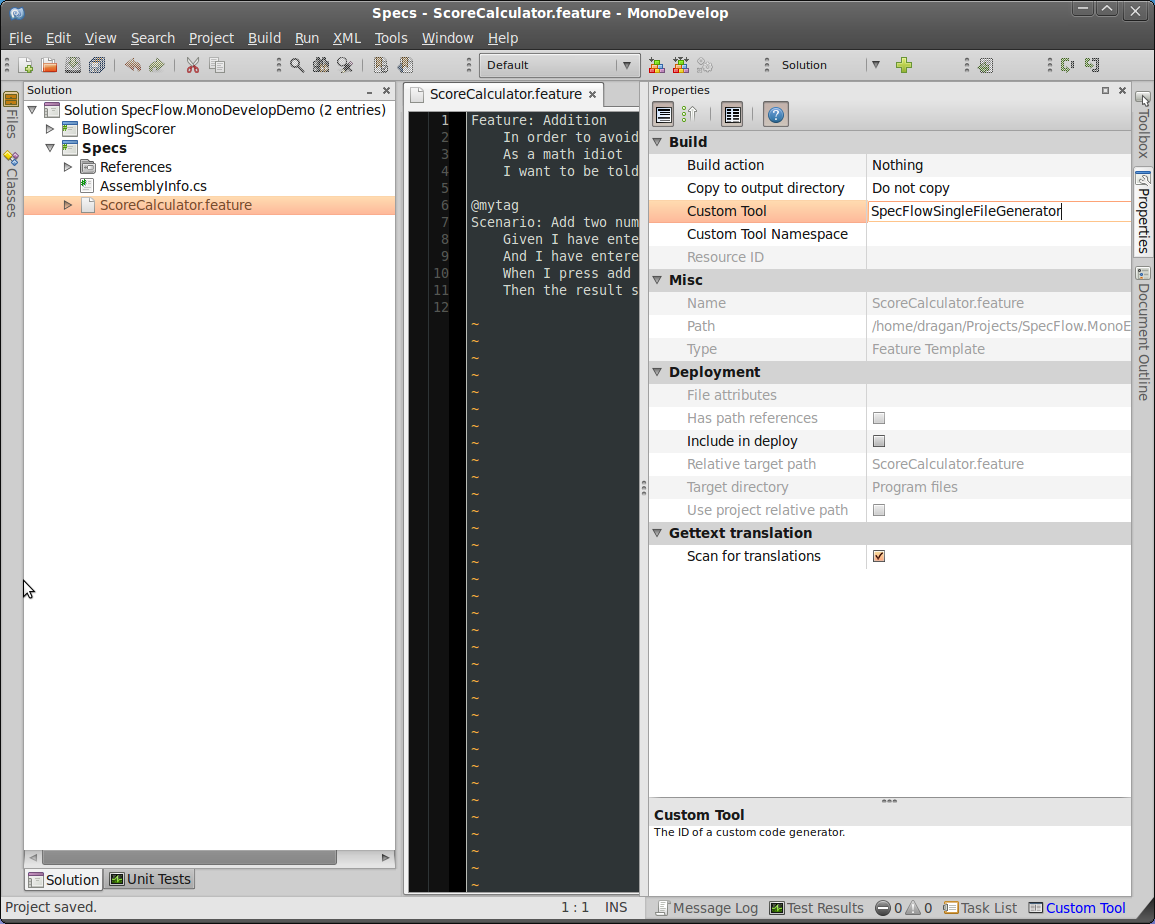
1. Right click the Specs project → Add → New File.
2. When the New File Dialog shows up, you will notice under SpecFlow, some new file templates that are provided by the Add-In.
3. Select the SpecFlow Feature template and give it a file name of ScoreCalculator.feature.



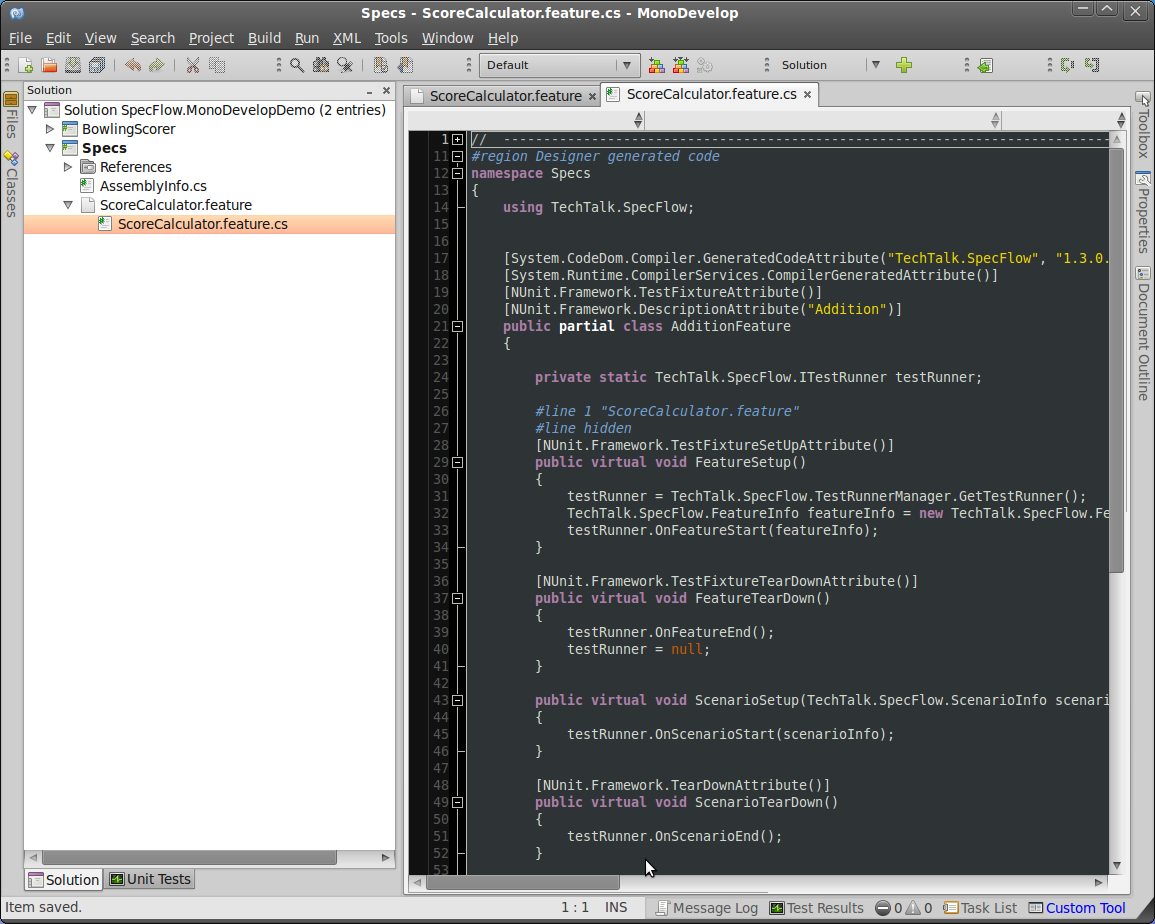
1. After clicking New, you are presented with the template auto filled with a default feature and scenario to remind you what the file looks like.



1. Every time you save this file, a custom tool is ran. To see what custom tool is run, select properties for the file. It should be populated with the SpecFlowSingleFileGenerator.



1. You will notice that it has generated a file for you, if select the arrow beside feature file and open the file that shows up.



1. You can run this as is in the NUnit Test Runner that is provided by MonoDevelop, but we actually want to create a feature, not some silly adder.