# Code conventions



> This information is based on http://csharpguidelines.codeplex.com, please read that. We have a few rules that are different, they are stated below

## Basic principles

Please keep in mind the following basic principles when writing code:

- The Principle of Least Surprise (or Astonishment), which means that you should choose a solution that does include any things people might not understand, or put on the wrong track.
- Keep It Simple Stupid (a.k.a. KISS), a funny way of saying that the simplest solution is more than sufficient.
- You Ain't Gonna Need It (a.k.a. YAGNI), which tells you to create a solution for the current problem rather than the ones you think will happen later on (since when can you predict the future?)
- Don't Repeat Yourself (a.k.a. DRY), which requires you to rigorously remove duplication in your code base

Regardless of the elegancy of somebody's solution, if it's too complex for the ordinary developer, or exposes unusual behavior, or tries to solve many possible future issues, it is very likely the wrong solution and needs redesign

## Naming conventions

1. Fields start with underscores, thus *_myField*
2. Constants and statics start with a capital, but are not fully capitalized, thus *Log*, *MyConstant*
3. Interfaces are always prefixed with an I, thus *IMyInterface*

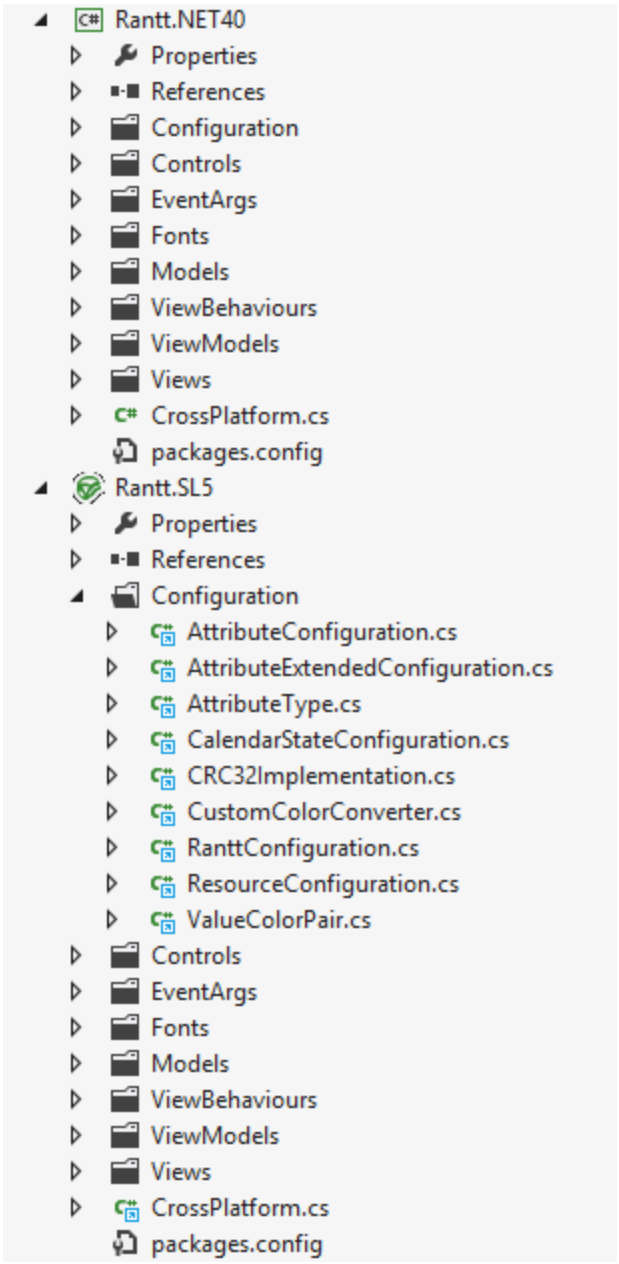
## Solution folders names and namespaces

The namespaces always follow the solution folders except for the following folders:

- Extensions
- Helpers
- Interfaces

## WPF, Silverlight and shared code

When using shared code for both WPF and Silverlight, the WPF is always leading and developed first. The reason for this is that it is always easy to add functionality to Silverlight which is not included by default. The files in the Silverlight project will be linked files to the WPF project. The naming conventions for the projects will have the same assembly name, but a different postfix (thus *NET40* and *SL5*). For a clarification, see the image below:


Rantt.NET40
Properties
References
Configuration
Controls
EventArgs
Fonts
Models
ViewBehaviours
ViewModels
Views
CrossPlatform.cs
packages.config
Rantt.SL5
Properties
References
Configuration
AttributeConfiguration.cs
AttributeExtendedConfiguration.cs
AttributeType.cs
CalendarStateConfiguration.cs
CRC32Implementation.cs
CustomColorConverter.cs
RanttConfiguration.cs
ResourceConfiguration.cs
ValueColorPair.cs
Controls
EventArgs
Fonts
Models
ViewBehaviours
ViewModels
Views
CrossPlatform.cs
packages.config