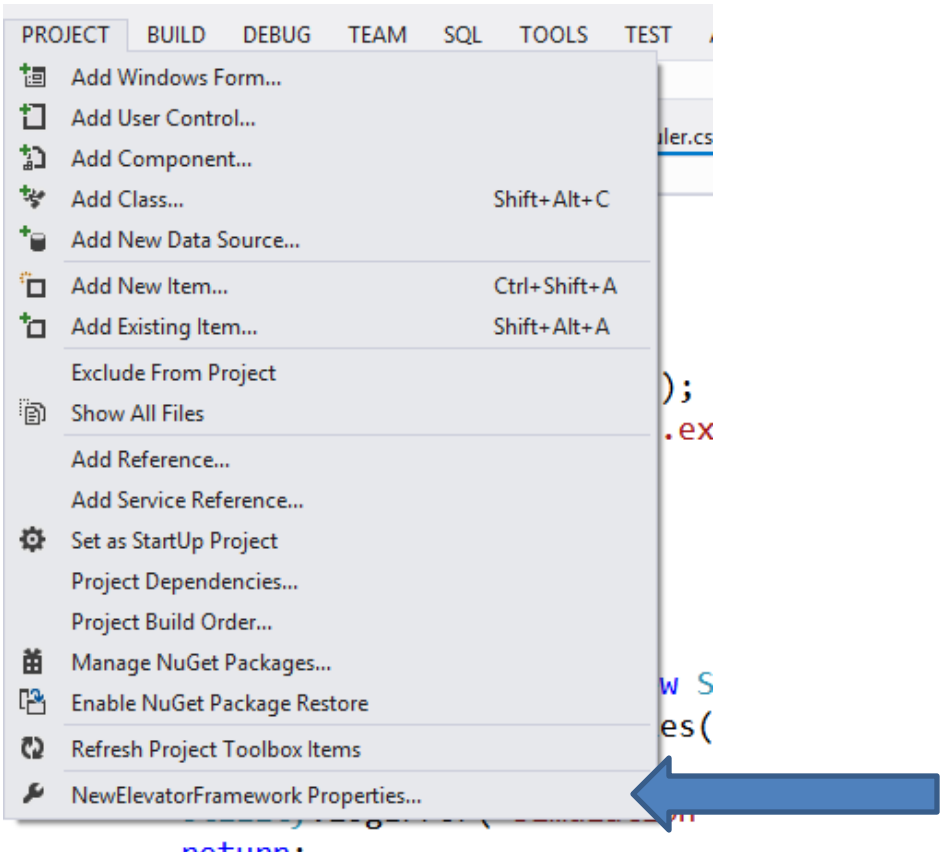
Suggestions

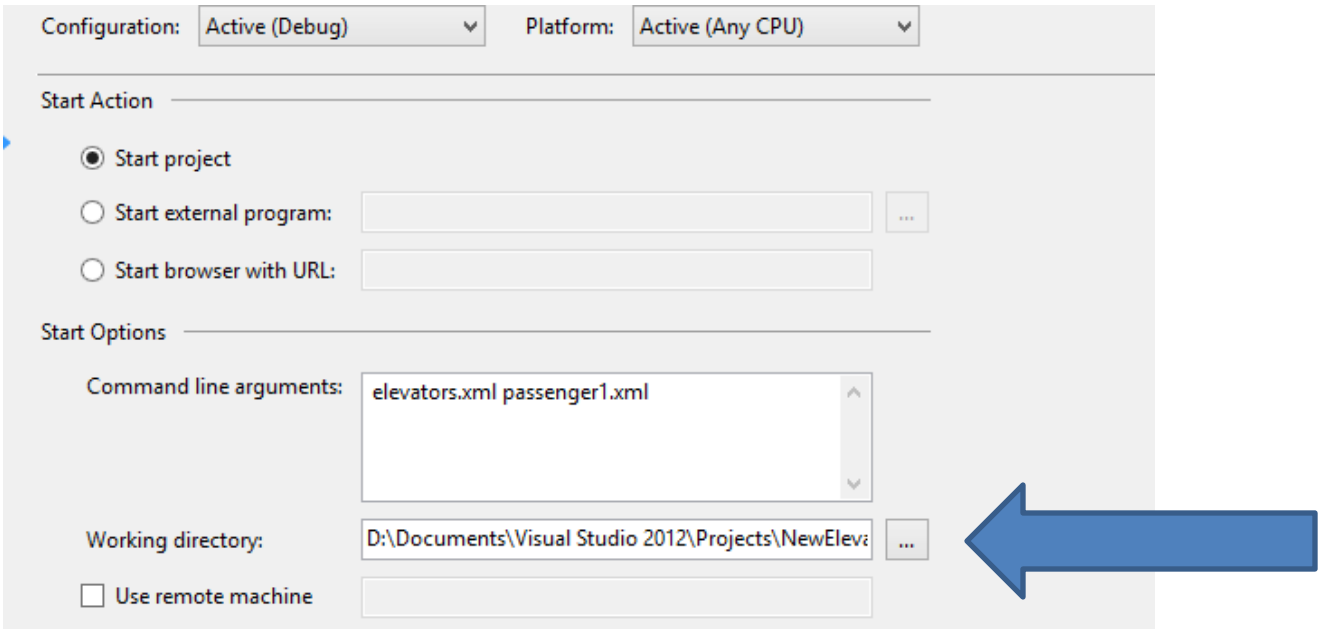
1、 Utility.cs 上面有两个宏定义，这两个宏定义控制着 debug 时控制台是否输出某些信息。

```
#define SHOW_DETAILS_IN_DEBUG
#define SHOW_WARNING_IN_DEBUG
```

在运行比较大的测试文件时，控制台的输出会极大的拖慢速度，可以通过注释掉这两个选项的办法来快速得到结果。

2、 开始运行程序时，需要电梯文件和乘客文件，debug 时，运行过后还有 log 文件，把文件拷到 debug、release 文件夹下是一个选择，还可以通过调整 project 的属性，指定一个 working directory，这样，就可以访问那个文件夹内的文件，生成的 log 文件也会在那个文件夹下：





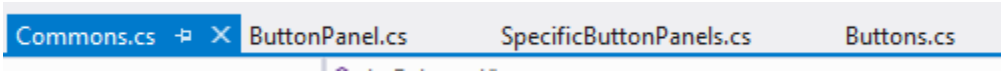
3、 如果你觉得代码太多，上下移动麻烦，可以在 vs2012 里按快捷键 ctrl+m+ctrl+o 将代码折叠后查看。

```
namespace NewElevatorFramework
{
    static class Utility
    {
        const int initialLogCounts = 10000;
        private static List<String> logRecordList = new List<string>(initialLogCounts);

        //methods ...
        public static void stopProgram()...
            //-------log record methods-------
        /// <summary> ...
        public static void log(string message)...

        public static void logWarning(string message) ...

        public static void logError(string message) ...

        private static void printInConsoleWithColor(string message,ConsoleColor color) ...

        /// <summary> ...
        public static Thread saveLogRecord(string fileName) ...

        //simulator
        public static double outputAnalysisResult(SimulateProgram simulator) ...

        //passengers
        public static int howManyPassengerInTotal(SimulateProgram simulator) ...

        public static int howManyPassengerArrived(SimulateProgram simulator) ...

        public static int howManyPassengerNonarrived(SimulateProgram simulator) ...
    }
}
```

4、 阅读代码的话，我建议先看 main 函数，比较易懂；再看 Utility 中的 log，logWarning ,logError 函数，因为这三个函数到处都用；

```
//methods ...
public static void stopProgram()...
    //-------log record methods-------
/// <summary> ...
public static void log(string message)...

public static void logWarning(string message) ...

public static void logError(string message) ...

private static void printInConsoleWithColor(string message,ConsoleColor color) ...

/// <summary> ...
public static Thread saveLogRecord(string fileName) ...
```

然后看看 Commons.cs 中的结构体，和 button，buttonPanel（包括 Buttons.cs,ButtonPanel.cs,SpecificButtonPanels.cs）；



接着看对于 passenger.cs, elevator.cs, Scheduler.cs 我的建议是：Passenger.cs 和 SimulateProgram.cs 中的 passenger action 系列函数结合着看， elevator 和 scheduler 结合着看。

```
void passengerTakeActions()...

void actionsInsideElevator(Passenger someOne)...
void actionsOutsideElevator(Passenger someOne)...
```

5、 调度算法添加到 Scheduler.cs 内的：

```
//-----------------Here add the schedule algorithm----------------
public void despatchQueriesToElev() ...
//generate bus schdule sequence
private int busGen(Direction direction,int lastFloor) ...
//----------------algorithm over-----------------------------------
```

可以把 despatchQueriesToElev 全部清空，自己重写。

6、 你可以在 scheduler 或是 elevator 中给它们添加优先队列，这都取决于调度算法。