TestSample Application Design Document

# Summary of TestSample Application

This application is used to merge files into single file for given Directory, it takes two parameters as input 1) Directory or Path with all the files with different extensions 2) Generic Outputfile Path where output of merged files get create into single file by extension

# Technical Design

Prerequisite : Software **Visual Studio 2015 Ultimate Edition** with .Net framework 4.5 or higher

Design is based on Startergy Pattern Where FileProcessor being injected to FileProcessManager Class

Each Different Type of File ( file having different extension) would be injected via Constructor of FileProcessManager. Each derived class of FileProcessor would have method called ProcessFile which would filtered out the Files from given Directory and read file and use ***Stringbuilder*** class to concatenate the content of result of each file.

For Special behaviour like ignoring lines with “@import” in CSS File behaviour would be implemented in respective FileProcessor class in this case CascadeStyleSheetFileProcessor class.

Common behaviour has been put into the FileProcessor Abstract Class like ReadFileData for better memory management and disposing the object like File Stream Reader ‘***using’*** block has been used. Most of the cases Stringbuilder has been used for string operations. Unity has been used in main Program to achieve DI ( Dependency Injection).

Configuration

<appSettings>

<add key ="JSOutputFileName" value ="\ALLJS.js"/>

<add key ="CSSOutputFileName" value ="\ALLCSS.css"/>

</appSettings>

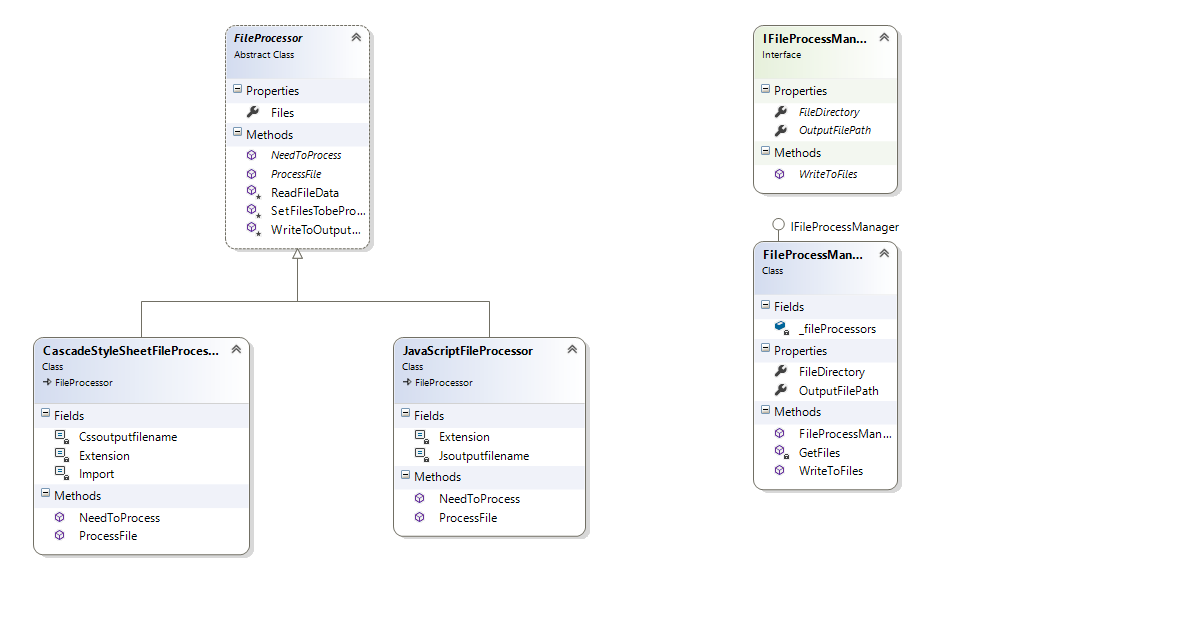
Output Single file can be configured from the above appSettings If new File extension comes for enhancement All Developer needs to do is implement / derive FileProcessor class for the new type of FileProcessor Add outputfilename in above configuration appsettings and Registered that type with its name in Unity Container.

## TestCases

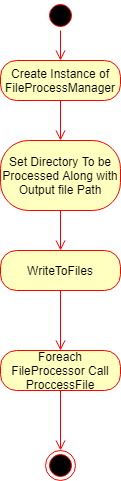
MSTest Unit Test Framework and MSFakes mocking framework are used for writing unit tests cases which has been written in different TestProject undert TestSample Solution. Each test class starts with name of the class end with “Test”.

Below are some of the UML Diagrams including activity and class diagram,

## Class Diagram



## Highlevel Activity Diagram



## Code Map

