LoadFileForChart Report

Mohammad Hossein Keshavaraz

august 2022

Contents

Chapter 1

Code Definition

In this class we are going to get a file from user and process its data for drawing purposes.

1.1 Input data and constructors

In this class we have implemented 2 constructors. Their difference is that one of them gets intput data for load buffer size and the other one selects the size automatically. The first input data is the file address. The second one is Resample Percentage which specifies the Resampling Rate in terms of percent. The third one is Example percent which specifies the ratio of the size of the example to the size of the input file as a percent. the last one is the size of the load buffer that will be used in loading and processing the data. If the last on remains empty the constructor will automatically initialize it.

Listing 1.1: First Constructor(without BufferSize)

```
1 public LoadFileForChart(string filePath,int reSamplePercentage,int
      examplePercent)
2         {
3             try
4             {
5                 FileInfo fileInfo = new FileInfo(filePath);
6                 this.FileName = fileInfo.FullName;
7                 this.FileSize = fileInfo.Length;
```

```
8                   this.ReSamplePercentage = reSamplePercentage;
9                   this.ExamplePercent = examplePercent;
10                  this.ExampleSize = FileSize * ExamplePercent /
                        100;
11                  this.LoadBufferSize = 1024 * 1024;
12                  this.ReSampleRate = 100/reSamplePercentage;
13              }
14              catch
15              {
16                  Console.WriteLine("couldnt get file data!");
17              }
18
19          }
```

Listing 1.2: Second Constructor(with BufferSize)

```
20  public LoadFileForChart(string filePath,int reSamplePercentage,int
        examplePercent)
21          {
22              try
23              {
24                  FileInfo fileInfo = new FileInfo(filePath);
25                  this.FileName = fileInfo.FullName;
26                  this.FileSize = fileInfo.Length;
27                  this.ReSamplePercentage = reSamplePercentage;
28                  this.ExamplePercent = examplePercent;
29                  this.ExampleSize = FileSize * ExamplePercent /
                        100;
30                  this.LoadBufferSize = 1024 * 1024;
31                  this.ReSampleRate = 100/reSamplePercentage;
32              }
33              catch
34              {
35                  Console.WriteLine("couldnt get file data!");
36              }
37
38          }
```

1.2 Properties

There are 7 properties that will be discussed in this class.

- FileName
  this property is used to store the name and address of the file.
- LoadBufferSize
  this property specifies the size of the load buffer.
- FileSize
  this property stores the size of the input file.
- ReSamplePercentage
  this property determines the percentage of the resampling to data ratio
- ReSampleRate
  this property determines the resampling rate. The constructor calculates it from ReSamplePercentage.
- ExamplePercent
  this property specifies the ratio of the size of the example to the size of the input file as a percent.
- ExampleSize
  this property stores the size of the example.

1.3 Methods

In this class we have different methods for getting the file and processing its data. Each class reads the file data as a different type(Int,Byte,Double).

- processDataByte
  This method reads bytes from file and process it. This will create an array of converted data for us. This method creates a binary reader object for reading data from file. It also create two byte arrays fileContent , convertedData. next it will use a (for) to read data from file and process it. In each cycle it will read data equal to loadbuffer size from the input file and store it in filecontent variable. Next it will Resample the data in filecontent variable using the resample rate prop and store the filtered data in convertedData variable. also it will use a

progress variable to calculate the advancement of the process and use a progress bar to show it.

- processDataDouble
  overall functionality of this method is similar to the previous one. the difference is that after creating the BinaryReader object there is not any method to read a block of data as double. so we will use a for loop to do so.

- processDataInt16
  this method works exactly like the processDataDouble. the only difference is their data types.

- processDataInt32
  this method works exactly like the processDataDouble. the only difference is their data types.

1.4 DataRelations

Figure (1.1) shows the relations between the input and processed data. As u can see, example percent prop determines How much of the output data of the resampling operation should be displayed in the example output.

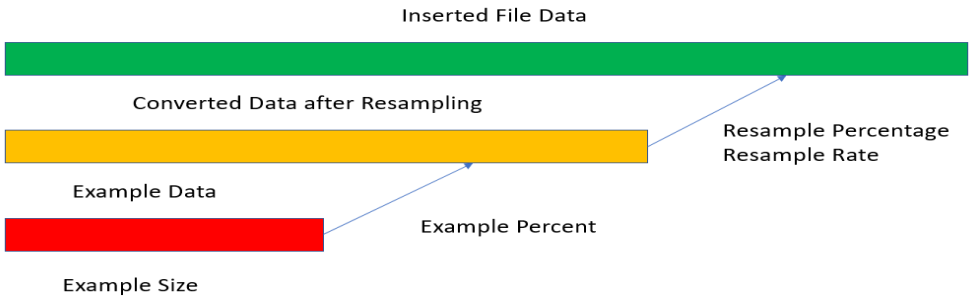

Figure 1.1: file and output data relations