Practical Guide 09

Read and Write Text Files

9.1 Open a File Using C#

When we work with programs, we should have the ability to manipulate files like text files, html files, Word docs, etc. Here, we will learn how to manipulate text files. First, we'll see how to open a text file in a C# program.

Exercise 1

1. Create a new Windows Form application and name it OpenFile.
2. First, you need to create text files that you need to open. Then, inside your project folder, create a new folder and name it TextFiles. Then create five text files and name them as "One"," Two"," Three"," Four", and "Five". Inside each text file, write some context that you can uniquely identify each.

   For example: In One.txt text file type 11111111 lines and in Two.txt file type 222222 lines.
3. Then go to Visual Studio and add a label, combo box, and a button, and create the interface as Figure 9.1. Change the Text property of the form to "Open File".

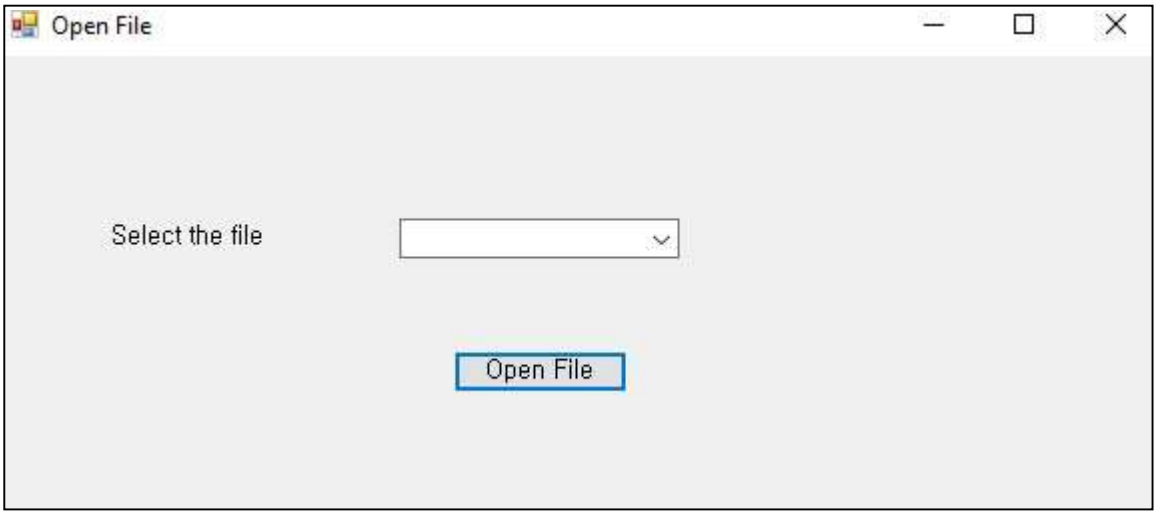

Figure 9.1 User Interface for open files

4. Click the combo box and go to the properties, and click the … in the Items property. Add the names of the text files you create in step 2.
5. Double click the "Open File" button and write the code below.

```
 private void button1_Click(object sender, EventArgs e)
 {
     try
     {
         string file1 = comboBox1.SelectedItem?.ToString();

         if (string.IsNullOrEmpty(file1))
         {
             MessageBox.Show("Please select a file to open");
             return;
         }

         string filePath = @"D:\Documents\Documents\Visual Programming\Class
Examples\TextFiles\" + file1 + ".txt";

         var psi = new ProcessStartInfo
         {
             FileName = filePath,
             UseShellExecute = true
         };

         Process.Start(psi);
     }
     Catch (Exception ex)
     {
         MessageBox.Show("Error opening file: " + ex.Message);
     }
    }
```

6. If the text files you need to open in the program are open, close them all and run the program. Open different files by selecting the file names in the combo box.

Note:

- `string file1 = comboBox1.SelectedItem?.ToString();`
  Gets the selected file name from the combo box safely.
- `if (string.IsNullOrEmpty(file1)) { ... }`
  Checks if no file is selected; shows a message and exits.
- `string filePath = ...`
  Builds a full path to the selected text file.
- `var psi = new ProcessStartInfo { ... }`
  Sets up process info to open the file with the default program.
- `Process.Start(psi);`
  Starts the process to open the file.
- `catch (Exception ex) { ... }`
  Handles errors by showing an error message.

9.2 Read and Write a File Using C#

In C# file operations, normally, streams are used to read and write to files. A stream is an additional layer created between an application and a file. The stream is used to ensure smooth read and write operations to the file.

Streams are normally used when reading data from large files. By using streams, the data from large files is broken down into small chunks and sent to the stream. These chunks of data can then be read from the application.

The reason for breaking it down into small chunks is because of the performance impact of reading a big file in one shot. If you were to read the data from, say, a 100 MB file at one shot, your application could just hang and become unstable. The best approach is then to use streams to break the file down into manageable chunks.

So, when a write operation is carried out on a file, the data to be written is first written to the stream. From the stream, the data is then written to the file. The same goes for the read operation. In the read operation, data is first transferred from the file to the stream. The data is then read from the application via the stream. Let's look at an example of how we can read and write using streams.

9.3 Read a File Using C#

Stream Reader

The stream reader is used to read data from a file using streams. The data from the file is first read into the stream. Thereafter, the application reads the data from the stream.

Exercise 2

1. Create a new project and name it FileReadWrite.
2. First, you need to create the text file to read. Create a folder inside the project and name it TextFiles. Inside the folder, create a text document and type the following text, and save the document.
   "Hi! Welcome to Visual Programming."
3. Add a Rich text box and two buttons to the form and create the user interface as given in Figure 9.2

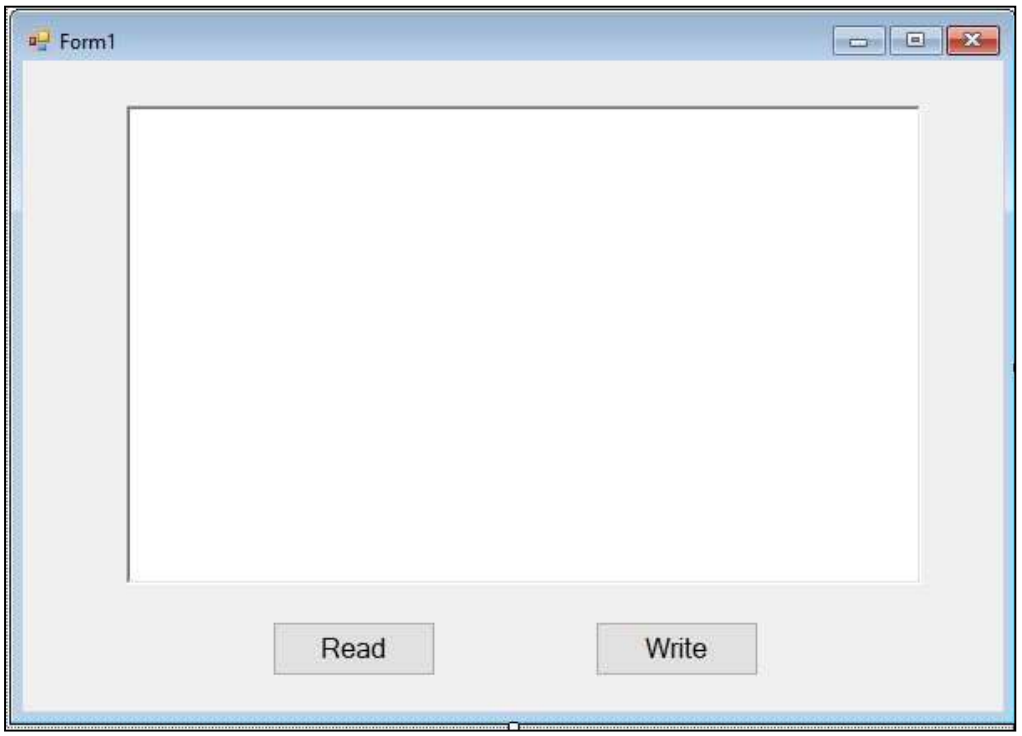

Figure 9.2 User Interface for read and write files

4. Double click the form and write the below code to the Form_Load event.

```
private void Form1_Load(object sender, EventArgs e)
{ richTextBox1.BackColor = Color.Beige;
   richTextBox1.ForeColor = Color.CornflowerBlue;

richTextBox1.Text = "Contents of the file will display here";
   richTextBox1.Font = new Font("Segoe Print", 9); }
```

5. Double click the read button and type the below code.

```
private void btn_Read_Click(object sender, EventArgs e)
        {
            String line;
            try
            { richTextBox1.Clear();
                richTextBox1.BackColor = Color.LightSeaGreen;
                richTextBox1.ForeColor = Color.WhiteSmoke;
                richTextBox1.Font = new Font("Times New Roman", 12);
                //Pass the file path and file name to the StreamReader constructor.
StreamReader sr = new StreamReader("D:\\Documents\\Documents\\Visual
Programming\\Class Examples\\TextFiles\\Read.txt");

                //Read the first line of text line
                = sr.ReadLine();

                //Continue to read until you reach the end of the
                file while (line != null)
                {
                    //write the file to the rich text box
                      richTextBox1.Text = richTextBox1.Text+line+Environment.NewLine;
                    //Read the next line line
                    = sr.ReadLine();
                }

                //close the file
                sr.Close();

            }
            catch (Exception e1)
            {
                MessageBox.Show("Exception: " + e1.Message);

            }

        }
```

6. Run the program by pressing F5 and click the read button. You can see that the text is appearing in the rich text box.

9.4 Write a File Using C#

Stream Writer

The stream writer is used to write data to a file using streams. The data from the application is first written into the stream. After that, the stream writes the data to the file.

Exercise 3

1. First, you need to create a text file inside the TextFiles folder to write. Create an empty text file and name it “Write”. 2. Double click the “Write” button and type the below code.

```
private void btn_write_Click(object sender, EventArgs e)
        { try
            {

               //Pass the filepath and filename to the StreamWriter Constructor
                StreamWriter sw = new StreamWriter("D:\\Documents\\Documents\\
                Visual Programming\\Class Examples\\TextFiles\\Write.txt");

                //Write a line of text
                sw.WriteLine("Hello!!!");

                //Write a second line of text sw.WriteLine("Good
                Morning...");

                //Write a third line of text
                sw.WriteLine("Welcome to Visual Programming.");

                //Close the file
                sw.Close();
            }
            catch (Exception e1)
            {
                MessageBox.Show("Exception: " +
        e1.Message); } }
```

3. Run the program and click the “Write” button. Make sure you have closed the “Write.txt” file.
4. After clicking the “Write” button, go and check whether the text in the text document has changed.

Exercise 4

1. Make modifications to the above form to open both text files. You can use any control or method according to preference.