|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CS 1400**  **Lab #25 Using the OpenFileDialog & SaveFileDialog**  **Version 1.0**  **Objectives:**  The objective of this lab is to learn how to use the **OpenFileDialog** and **SaveFileDialog** and read data from a file and display it in a ListBox.  **The Interface**  Create a new ***Windows Forms Application***. Add a ***MenuStrip*** to the top of the Form and include the standard ***Exit*** and ***About*** menu items. Add a “Read File” Button and a “Write File” Button. Then place a ***TextBox*** control someplace at the top of the Form. Add a ***multiple line TextBox*** to the right of the ***Read/Write Buttons***. When you are done, your interface should look something like the images below Fig. 1 and 2:    Figure 1File Read/Write Form  Figure Menu Items  **Using OpenFileDialog**   1. Click on the ***ReadFile*** Button. This will open up the code editor and create a method to handle the event that will get generated when a user clicks on ***“Read File” Button***. Here you will add the code to create a ***OpenFileDialog*** and read some data from the file. When you are done, your event handler code should look something like this:   **private** **void** **BtnReadFile\_Click**(**object** **sender**, EventArgs **e**)  {  **const** **int** **TASIZE** **=** 2;  **TxtDisplayData.Clear**();  **TxtFileName.Text** **=** **"Filename will be displayed here!"**;  **openDlg** **=** **new** OpenFileDialog();  **openDlg.InitialDirectory** **=** Environment**.GetFolderPath**(Environment**.**SpecialFolder**.Personal**);  **openDlg.Filter** **=** **"All Files (\*.\*)|\*.\*|Text Files (\*.txt)|\*.txt"**;  **openDlg.FilterIndex** **=** 2;  **string** **data** **=** **null**;  **string** **tdata** **=** **null**;  **string**[] **adata** **=** **new** **string**[**TASIZE**];  **try**  {  **if** (**openDlg.ShowDialog**() **==** DialogResult**.OK**)  {  **if** ((**fs** **=** (FileStream)(**openDlg.OpenFile**())) **!=** **null**)  {  **strRdr** **=** **new** StreamReader(**fs**);  **TxtFileName.Text** **=** **openDlg.FileName**;  **for** (**int** **idx** **=** 0; **idx** **<** **grades.Length**; **idx++**)  {  **data** **=** **strRdr.ReadLine**();  **if** (**data** **==** **null || data = ""**)  {  **grades**[**idx**] **=** **-**1.0;  **return**;  }  **else**  {  **adata** **=** **data.Split**();  **grades**[**idx**] **=** **double.Parse**(**adata**[1]);  **tdata** **=** **string.Format**(**"{0}\t{1:F2}\r\n"**, **adata**[0], **grades**[**idx**]);  **TxtDisplayData.AppendText**(**tdata**);  }  }  **if** (**strRdr** **!=** **null**)  **strRdr.Close**();  **fs.Close**();  }  **else**  **throw** **new** IOException(**"File failed to open"**);  }  **else**  **throw** **new** Exception(**"File Open Cancelled"**);  }  **catch** (IOException **ioexp**)  {  MessageBox**.Show**(**"File "** **+** **saveDlg.FileName** **+** **ioexp.Message**, **"File Open Failed"**,  MessageBoxButtons**.OK**, MessageBoxIcon**.Error**);  }  **catch** (Exception **exp**)  {  MessageBox**.Show**(**exp.Message**, **"Open Cancelled"**, MessageBoxButtons**.OK**,  MessageBoxIcon**.Information**);  }  }**//End BtnReadFile\_Click( )**   1. Be sure to add a ***using System.IO***; statement at the top of the file. The code above does the following: 2. 1. Creates a ***OpenFileDialog*** object and returns it’s reference.   2. Stores the initial directory in the **openDlg.InitialDirectory** property.   1. 3. Initializes the Filter for the ***OpenFileDialog*** to the text string which 2. defines the file extensions to show in the file dialog. 3. 4. Displays the ***OpenFileDialog*** and checks the return value to make sure that this operation worked. 4. 5. Opens the file and assigns the reference to the stream object to ***strRdr***. 5. 6. If the file did not open for some reason the reference will be ***null***. 6. 7. Reads each line of data from the file and converts it to an int and stores it the grades array and displays it in the ***TxtDisplayData*** object. 7. 10. Build, test and refactor your program to make sure it meets the specifications described in this lab.   **Using SaveFileDialog**  Click on the ***“Write File”*** ***Button***. This will open up the code editor and create a method to handle the event that will get generated when a user clicks on Open. Here you will add the code to create a ***SaveFileDialog*** and write data to the file. The example Form shows data values I used when writing the program. When you are done, your event handler code should look something like this:  **private** **void** **BtnWriteFile\_Click**(**object** **sender**, EventArgs **e**)  {  **BtnReadFile.Enabled** **=** **true**;  **TxtDisplayData.Clear**();  **TxtFileStatus.Text** **=** **"Filename or other status will be displayed here!"**;  **saveDlg** **=** **new** SaveFileDialog();  **saveDlg.InitialDirectory** **=** Environment**.GetFolderPath**(Environment**.**SpecialFolder**.Personal**);  **saveDlg.Filter** **=** **"All Files (\*.\*)|\*.\*|Text Files (\*.txt)|\*.txt"**;  **saveDlg.FilterIndex** **=** 2;  **try**  {  **if** ((**dlgResult** **=** **saveDlg.ShowDialog**()) **==** DialogResult**.OK**)  {  **if** ((**\_fs** **=** (FileStream)**saveDlg.OpenFile**()) **!=** **null**)  {  **\_strmWrt** **=** **new** StreamWriter(**\_fs**);  **TxtFileStatus.Text** **=** **saveDlg.FileName**;  **for** (**int** **idx** **=** 0; **idx** **<** **ASIZE**; **idx++**)  {  **\_strmWrt.WriteLine**(**msg+idx** **+** **"\t"** **+** **data**);  **TxtDisplayData.AppendText**(**string.Format**(**"{0}\t{1:F2}\r\n"**, **msg+idx**, **data**));  **data** **+=** **inc**;  }  }  **else** **if**(**dlgResult==**DialogResult**.Cancel**)  **throw** **new** IOException(**"File Open \"Cancelled!\""**);  }  **else**  **throw** **new** Exception(**"Unknown DialogResult!"**);  }  **catch** (IOException **ioexp**)  {  MessageBox**.Show**(**"File "** **+** **saveDlg.FileName** **+** **ioexp.Message**,  **"File Open Failed"**,MessageBoxButtons**.OK**, MessageBoxIcon**.Error**);  **BtnReadFile.Enabled** **=** **false**;  }  **catch** (Exception **exp**)  {  MessageBox**.Show**(**exp.Message**, **"Open Cancelled"**,  MessageBoxButtons**.OK**,MessageBoxIcon**.Information**);  **BtnReadFile.Enabled** **=** **false**;  }  **finally**  {  **openDlg** **=** **null**;  **saveDlg** **=** **null**;  **if** (**\_strmWrt** **!=** **null**)  {  **\_strmWrt.Close**();  **\_strmWrt** **=** **null**;  }  **if**(**\_fs!=null**)  {  **\_fs.Close**();  **\_fs** **=** **null**;  }  }  }   1. Be sure to add a ***using System.IO***; statement at the top of the file. The code above does the following: 2. 1. Creates a ***SaveFileDialog*** object and returns it’s reference.   2. Stores the initial directory in the **saveDlg.InitialDirectory** property.   1. 3. Initializes the Filter for the ***SaveFileDialog*** to the text string which 2. defines the file extensions to show in the file dialog. 3. 4. Displays the ***SaveFileDialog*** and checks the return value to make sure that this operation worked. 4. 5. Opens the file and assigns the reference to the stream object to ***strWtr***. 5. 6. If the file did not open for some reason the reference will be ***null***. 6. 7. Writes each line of data from to the file and converts the double values to strings with WriteLine and displays it in the ***TxtDisplayData*** object. 7. 10. Build, test and refactor your program to make sure it meets the specifications described in this lab.   Add the following:   * 1. Read File Button   2. Code in the BtnReadFile\_Click(…) handler to read the file written.   3. Display the formatted contents of the data read.   **File(s) to Submit:**  Place your complete project folder in a zip file and name the zip file Lab\_25\_your-initials\_V1.0.zip. For example, I would name my file Lab\_25\_DAF\_V1.0.zip. Submit this assignment as Lab #25 on Canvas.   |  |  |  | | --- | --- | --- | |  | **Grading Guidelines** |  | | # | Program | C(correct)  X(incorrect) | | 1 | Meets & works to specifications | 6 points | | 2 | Error Free, elegant & efficient | 4 points | | 3 | Pseudo-Code | -3 points | | 4 | Style Guidelines | -2 points | | 5 | Header File(s) & Formatting | -2 points | | 6 | Source Files(s) & Formatting | -1 points | | 7 | Project Prolog | -1 points | | 8 | Function Prologs | -1 points | | 9 | Zip Filename | -1 points | | 10 | Lab & Project Names | -1 points | | 11 | Zip File is invalid or will not unzip | Lab = 0 pts | |  | Total Points | 10 | 0-9 | |