

# PDF Combiner

A simple Windows Forms app to combine multiple PDF files in a single PDF file

# Table of Contents

|                                  |           |
|----------------------------------|-----------|
| <b>Symbol Reference</b>          | <b>1</b>  |
| <b>PdfCombiner Namespace</b>     | <b>1</b>  |
| PdfCombiner.Properties Namespace | 1         |
| Classes                          | 1         |
| Resources Class                  | 1         |
| Settings Class                   | 3         |
| Classes                          | 4         |
| BaseException Class              | 4         |
| BaseException Constructor        | 4         |
| ExceptionExtensions Class        | 5         |
| ExceptionExtensions Methods      | 6         |
| FrmMain Class                    | 6         |
| FrmMain.FrmMain Constructor      | 8         |
| FrmMain Fields                   | 8         |
| FrmMain Methods                  | 11        |
| Program Class                    | 26        |
| Program Methods                  | 27        |
| <b>Files</b>                     | <b>27</b> |
| AssemblyInfo.cs                  | 27        |
| BaseException.cs                 | 28        |
| ExceptionExtensions.cs           | 29        |
| FrmMain.cs                       | 29        |
| FrmMain.Designer.cs              | 38        |
| PdfCombiner.csproj               | 43        |
| PdfCombiner.sln                  | 43        |
| Program.cs                       | 43        |
| Resources.Designer.cs            | 44        |
| Settings.Designer.cs             | 45        |
| <b>Index</b>                     | <b>a</b>  |

# 1 Symbol Reference





## 1.1 PdfCombiner Namespace

This is namespace PdfCombiner.

### Namespaces

| Name                                      | Description                               |
|---|---|
| Properties ( <a href="#">see page 1</a> ) | This is namespace PdfCombiner.Properties. |



### Classes

|  | Name   | Description  |
|--|--|--|
|   | BaseException ( <a href="#">see page 4</a> )       | This function is used to take exceptions which can be thrown in PDF Combiner App |
|   | ExceptionExtensions ( <a href="#">see page 5</a> ) | This is class PdfCombiner.ExceptionExtensions.                                   |
|   | FrmMain ( <a href="#">see page 6</a> )             | This is class PdfCombiner.FrmMain.   |
|  | Program ( <a href="#">see page 26</a> )            | This is class PdfCombiner.Program.   |

### 1.1.1 PdfCombiner.Properties Namespace

This is namespace PdfCombiner.Properties.



### Classes

|   | Name                                     | Description   |
|---|--|---|
|  | Resources ( <a href="#">see page 1</a> ) | A strongly-typed resource class, for looking up localized strings, etc. |
|  | Settings ( <a href="#">see page 3</a> )  | This is class PdfCombiner.Properties.Settings.                          |

#### 1.1.1.1 Classes

The following table lists classes in this documentation.

### Classes

|   | Name                                     | Description   |
|---|--|---|
|  | Resources ( <a href="#">see page 1</a> ) | A strongly-typed resource class, for looking up localized strings, etc. |
|  | Settings ( <a href="#">see page 3</a> )  | This is class PdfCombiner.Properties.Settings.                          |

#### 1.1.1.1.1 Resources Class

A strongly-typed resource class, for looking up localized strings, etc.

### Class Hierarchy

PdfCombiner.Properties.Resources

**C#**

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Resources.Tools.StronglyTypedResourceBuilder",  
"15.0.0.0")]  
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]  
[global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]  
internal class Resources;
```


**File**

Resources.Designer.cs (see page 44)



**Description**

This class was auto-generated by the StronglyTypedResourceBuilder class via a tool like ResGen or Visual Studio. To add or remove a member, edit your .ResX file then rerun ResGen with the /str option, or rebuild your VS project.




**Methods**

|   | Name                   | Description                                     |
|---|------------------------|---|
|  | Resources (see page 2) | This is Resources, a member of class Resources. |

**Resources Fields**

|   | Name                         | Description   |
|---|------------------------------|---|
|  | resourceCulture (see page 2) | This is resourceCulture, a member of class Resources. |
|  | resourceMan (see page 3)     | This is resourceMan, a member of class Resources.     |

**Resources Properties**

|   | Name                         | Description   |
|---|------------------------------|---|
|  | AppTitle (see page 3)        | Looks up a localized string similar to PDF Birlestirici.  |
|  | Culture (see page 3)         | Overrides the current thread's CurrentUICulture property for all resource lookups using this strongly typed resource class. |
|  | ResourceManager (see page 3) | Returns the cached ResourceManager instance used by this class.   |

**1.1.1.1.1.1 Resources.Resources Constructor****C#**

```
[global::System.Diagnostics.CodeAnalysis.SuppressMessageAttribute("Microsoft.Performance",  
"CA1811:AvoidUncalledPrivateCode")]  
internal Resources();
```

**Description**

This is Resources, a member of class Resources.

**Body Source**

```
1:  
[global::System.Diagnostics.CodeAnalysis.SuppressMessageAttribute("Microsoft.Performance",  
"CA1811:AvoidUncalledPrivateCode")]  
2: internal Resources() {  
3: }
```

**1.1.1.1.1.2 Resources Fields****1.1.1.1.1.2.1 Resources.resourceCulture Field****C#**

```
private static global::System.Globalization.CultureInfo resourceCulture;
```

**Description**

This is resourceCulture, a member of class Resources.

**1.1.1.1.1.2.2 Resources.resourceMan Field****C#**

```
private static global::System.Resources.ResourceManager resourceMan;
```

**Description**

This is resourceMan, a member of class Resources.

**1.1.1.1.1.3 Resources Properties****1.1.1.1.1.3.1 Resources.AppTitle Property**

Looks up a localized string similar to PDF Birlestirici.

**C#**

```
internal static string AppTitle;
```

**1.1.1.1.1.3.2 Resources.Culture Property**

Overrides the current thread's CurrentUICulture property for all resource lookups using this strongly typed resource class.

**C#**

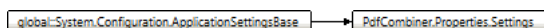
```
[global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableState.Advanced)]
internal static global::System.Globalization.CultureInfo Culture;
```

**1.1.1.1.1.3.3 Resources.ResourceManager Property**

Returns the cached ResourceManager instance used by this class.

**C#**

```
[global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableState.Advanced)]
internal static global::System.Resources.ResourceManager ResourceManager;
```

**1.1.1.1.2 Settings Class****Class Hierarchy****C#**

```
[global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.VisualStudio.Editors.SettingsDesigner.SettingsSingleFileGenerator",
"11.0.0.0")]
internal sealed class Settings : global::System.Configuration.ApplicationSettingsBase;
```


**File**

Settings.Designer.cs (📄 see page 45)

**Description**

This is class PdfCombiner.Properties.Settings.

**Settings Fields**

|   | Name   | Description  |
|---|--|--|
|  | defaultInstance ( <a href="#">see page 4</a> ) | This is defaultInstance, a member of class Settings. |

**1.1.1.1.2.1 Settings Fields****1.1.1.1.2.1.1 Settings.defaultInstance Field****C#**

```
private static Settings defaultInstance =  
((Settings)(global::System.Configuration.ApplicationSettingsBase.Synchronized(new  
Settings())));
```





**Description**

This is defaultInstance, a member of class Settings.

## 1.1.2 Classes

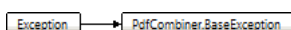
The following table lists classes in this documentation.

**Classes**

|   | Name   | Description  |
|---|--|--|
|   | BaseException ( <a href="#">see page 4</a> )       | This function is used to take exceptions which can be thrown in PDF Combiner App |
|  | ExceptionExtensions ( <a href="#">see page 5</a> ) | This is class PdfCombiner.ExceptionExtensions.                                   |
|  | FrmMain ( <a href="#">see page 6</a> )             | This is class PdfCombiner.FrmMain.   |
|  | Program ( <a href="#">see page 26</a> )            | This is class PdfCombiner.Program.   |

**1.1.2.1 BaseException Class**

This function is used to take exceptions which can be thrown in PDF Combiner App


**Class Hierarchy****C#**

```
[Serializable]  
public abstract class BaseException : Exception;
```

**File**

BaseException.cs ([see page 28](#))

**Methods**

|   | Name   | Description   |
|---|--|---|
|  | BaseException ( <a href="#">see page 5</a> ) | This is BaseException, a member of class BaseException. |

**1.1.2.1.1 BaseException Constructor**

### 1.1.2.1.1.1 BaseException.BaseException Constructor ()

C#

```
protected BaseException();
```

#### Description

This is BaseException, a member of class BaseException.

#### Body Source

```
1: protected BaseException()  
2: {  
3: }
```

### 1.1.2.1.1.2 BaseException.BaseException Constructor (string)

C#

```
protected BaseException(string message);
```

#### Description

This is BaseException, a member of class BaseException.

#### Body Source

```
1: protected BaseException(string message) : base(message)  
2: {  
3: }
```

### 1.1.2.1.1.3 BaseException.BaseException Constructor (string, Exception)

C#

```
protected BaseException(string message, Exception innerException);
```

#### Description

This is BaseException, a member of class BaseException.

#### Body Source

```
1: protected BaseException(string message, Exception innerException) : base(message,  
innerException)  
2: {  
3: }
```

## 1.1.2.2 ExceptionExtensions Class

#### Class Hierarchy

PdfCombiner.ExceptionExtensions

C#

```
public static class ExceptionExtensions;
```



#### File

ExceptionExtensions.cs ([🔗](#) see page 29)

#### Description

This is class PdfCombiner.ExceptionExtensions.

**ExceptionExtensions Methods**

|   | Name   | Description   |
|---|--|---|
|  | GetAllMessages (  see page 6) | This method is used to take all messages with stack trace info in exception |

**1.1.2.2.1 ExceptionExtensions Methods****1.1.2.2.1.1 ExceptionExtensions.GetAllMessages Method**

This method is used to take all messages with stack trace info in exception

**C#**

```
public static string GetAllMessages(this Exception exception);
```

**Parameters**

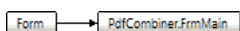
| Parameters               | Description |
|--------------------------|-------------|
| this Exception exception | Exception   |

**Returns**

Detailed Error Message


**Body Source**

```
1: public static string GetAllMessages(this Exception exception)
2: {
3:     var messages = exception.Message;
4:     if (exception.InnerException != null)
5:         messages += " " + exception.InnerException.GetAllMessages();
6:     if (!string.IsNullOrEmpty(exception.StackTrace))
7:         messages += " " + exception.StackTrace;
8:     if (!string.IsNullOrEmpty(exception.Source))
9:         messages += " " + exception.Source;
10:    return messages;
11: }
```

**1.1.2.3 FrmMain Class****Class Hierarchy****C#**

```
public class FrmMain : Form;
```

**File**

FrmMain.Designer.cs ( see page 38)

**Description**


















This is class PdfCombiner.FrmMain.

**Methods**










|   | Name  | Description                                 |
|---|---|---|
|  | FrmMain (  see page 8) | This is FrmMain, a member of class FrmMain. |



**FrmMain Fields**

|   | Name  | Description   |
|---|---|---|
|  | btnAddFile ( <a href="#">see page 8</a> )                     | This is btnAddFile, a member of class FrmMain.                    |
|  | btnAddFolder ( <a href="#">see page 9</a> )                   | This is btnAddFolder, a member of class FrmMain.                  |
|  | btnClearList ( <a href="#">see page 9</a> )                   | This is btnClearList, a member of class FrmMain.                  |
|  | btnCombineITextSharp ( <a href="#">see page 9</a> )           | This is btnCombineITextSharp, a member of class FrmMain.          |
|  | btnCombinePdfSharp ( <a href="#">see page 9</a> )             | This is btnCombinePdfSharp, a member of class FrmMain.            |
|  | cmbLanguage ( <a href="#">see page 9</a> )                    | This is cmbLanguage, a member of class FrmMain.                   |
|  | components ( <a href="#">see page 9</a> )                     | Required designer variable.                                       |
|  | lbFiles ( <a href="#">see page 9</a> )                        | This is lbFiles, a member of class FrmMain.                       |
|  | lblDetails ( <a href="#">see page 10</a> )                    | This is lblDetails, a member of class FrmMain.                    |
|  | menuItemDelete ( <a href="#">see page 10</a> )                | This is menuItemDelete, a member of class FrmMain.                |
|  | menuItemOrderByNameAscending ( <a href="#">see page 10</a> )  | This is menuItemOrderByNameAscending, a member of class FrmMain.  |
|  | menuItemOrderByNameDescending ( <a href="#">see page 10</a> ) | This is menuItemOrderByNameDescending, a member of class FrmMain. |
|  | menuItemOrderByPathAscending ( <a href="#">see page 10</a> )  | This is menuItemOrderByPathAscending, a member of class FrmMain.  |
|  | menuItemOrderByPathDescending ( <a href="#">see page 10</a> ) | This is menuItemOrderByPathDescending, a member of class FrmMain. |
|  | menuStrip ( <a href="#">see page 10</a> )                     | This is menuStrip, a member of class FrmMain.                     |
|  | pbFiles ( <a href="#">see page 11</a> )                       | This is pbFiles, a member of class FrmMain.                       |
|  | resource ( <a href="#">see page 11</a> )                      | This is resource, a member of class FrmMain.                      |

**FrmMain Methods**

|   | Name   | Description  |
|---|--|--|
|  | btnAddFile_Click ( <a href="#">see page 11</a> )                 | It is used to add single or multiple PDF files to combine When you choose file or files The listbox in the form will be filled with the name of the files Which you choose to combine  |
|  | btnAddFolder_Click ( <a href="#">see page 12</a> )               | This function is used to add PDF files in a folder which you choose in Folder Dialog recursively When you use it the added files will be seen in listbox   |
|  | btnClearList_Click ( <a href="#">see page 12</a> )               | This function is used to clear the file list in listbox  |
|  | btnCombineITextSharp_Click ( <a href="#">see page 13</a> )       | This function is used to take list of PDF files in listbox, Combine them with a single file in location which you choose in folder dialog And you can see the progress in progress bar It uses iTextSharp Nuget Package                                      |
|  | btnCombinePdfSharp_Click ( <a href="#">see page 14</a> )         | This function is used to take list of PDF files in listbox, Combine them with a single file in location which you choose in folder dialog And you can see the progress in progress bar It uses PdfSharp Nuget Package  |
|  | cmbLanguage_SelectedIndexChanged ( <a href="#">see page 15</a> ) | This function is used to set the resource ( <a href="#">see page 11</a> ) file And with this, you can see form elements, info messages etc. with the selected resource ( <a href="#">see page 11</a> ) file contents. That is used to multi language support |
|  | DeleteFilesFromListBox ( <a href="#">see page 16</a> )           | This method is used to Delete selected files from ListBox This is used different ways in this app So we created this as a method   |
|  | Dispose ( <a href="#">see page 16</a> )                          | Clean up any resources being used.   |
|  | FormMembersNameInitialization ( <a href="#">see page 17</a> )    | This method is used to change text of form members by the Resource file which is selected  |

|   |   |  |
|---|---|--|
|    | FrmMain_FormClosed (see page 17)                  | This function is used to terminate application   |
|    | FrmMain_Shown (see page 18)                       | This function is used to assign menu to listbox When form is shown   |
|    | InitializeComponent (see page 18)                 | Required method for Designer support - do not modify the contents of this method with the code editor.   |
|       | InitializeFileDialog (see page 21)                | This function is used to set options of file dialog like Filter, start path etc.   |
|    | lbFiles_DragDrop (see page 22)                    | This is used to drag and drop an item in listbox Which is used to order books and combine them with that order   |
|    | lbFiles_DragOver (see page 22)                    | This is used to give effect while dragging and dropping an item in listbox   |
|    | lbFiles_KeyDown (see page 22)                     | This function is used to delete files in listbox which are chosen If you press the Delete button in your keyboard The selected files will be deleted from the list |
|    | lbFiles_MouseDown (see page 23)                   | This is used to control mouse actions and if that is not right click Start to drag and drop item in listbox  |
|    | menuItemDelete_Click (see page 23)                | This is used to delete selected items in listbox When menu item Delete is clicked and opened dialog  |
|    | menuItemOrderByNameAscending_Click (see page 24)  | This method is used to order elements in listbox ascending by file name  |
|    | menuItemOrderByNameDescending_Click (see page 24) | This method is used to order elements in listbox descending by file name   |
|    | menuItemOrderByPathAscending_Click (see page 24)  | This method is used to order elements in listbox ascending by full path  |
|    | menuItemOrderByPathDescending_Click (see page 25) | This method is used to order elements in listbox descending by full path   |
|       | SortItemsByName (see page 25)                     | This method is used to order items by file name in listbox And return them with the given order as parametre   |
|   | SortItemsByPath (see page 26)                     | This method is used to order items by full path in listbox And return them with the given order as parametre   |

### 1.1.2.3.1 FrmMain.FrmMain Constructor

C#

```
public FrmMain();
```

#### Description

This is FrmMain, a member of class FrmMain.

#### Body Source

```
1: public FrmMain()
2: {
3:     InitializeComponent();
4: }
```

### 1.1.2.3.2 FrmMain Fields

#### 1.1.2.3.2.1 FrmMain.btnAddFile Field

C#

```
private System.Windows.Forms.Button btnAddFile;
```

#### Description

This is btnAddFile, a member of class FrmMain.

### 1.1.2.3.2.2 FrmMain.btnAddFolder Field

**C#**

```
private System.Windows.Forms.Button btnAddFolder;
```

**Description**

This is btnAddFolder, a member of class FrmMain.

### 1.1.2.3.2.3 FrmMain.btnClearList Field

**C#**

```
private System.Windows.Forms.Button btnClearList;
```

**Description**

This is btnClearList, a member of class FrmMain.

### 1.1.2.3.2.4 FrmMain.btnCombineITextSharp Field

**C#**

```
private System.Windows.Forms.Button btnCombineITextSharp;
```

**Description**

This is btnCombineITextSharp, a member of class FrmMain.

### 1.1.2.3.2.5 FrmMain.btnCombinePdfSharp Field

**C#**

```
private System.Windows.Forms.Button btnCombinePdfSharp;
```

**Description**

This is btnCombinePdfSharp, a member of class FrmMain.

### 1.1.2.3.2.6 FrmMain.cmbLanguage Field

**C#**

```
public System.Windows.Forms.ComboBox cmbLanguage;
```

**Description**

This is cmbLanguage, a member of class FrmMain.

### 1.1.2.3.2.7 FrmMain.components Field

Required designer variable.

**C#**

```
private System.ComponentModel.IContainer components;
```

### 1.1.2.3.2.8 FrmMain.lbFiles Field

**C#**

```
private System.Windows.Forms.ListBox lbFiles;
```

**Description**

This is lbFiles, a member of class FrmMain.

### 1.1.2.3.2.9 FrmMain.lblDetails Field

C#

```
private System.Windows.Forms.Label lblDetails;
```

#### Description

This is lblDetails, a member of class FrmMain.

### 1.1.2.3.2.10 FrmMain.menuItemDelete Field

C#

```
private System.Windows.Forms.ToolStripItem menuItemDelete;
```

#### Description

This is menuItemDelete, a member of class FrmMain.

### 1.1.2.3.2.11 FrmMain.menuItemOrderByNameAscending Field

C#

```
private System.Windows.Forms.ToolStripItem menuItemOrderByNameAscending;
```

#### Description

This is menuItemOrderByNameAscending, a member of class FrmMain.

### 1.1.2.3.2.12 FrmMain.menuItemOrderByNameDescending Field

C#

```
private System.Windows.Forms.ToolStripItem menuItemOrderByNameDescending;
```

#### Description

This is menuItemOrderByNameDescending, a member of class FrmMain.

### 1.1.2.3.2.13 FrmMain.menuItemOrderByPathAscending Field

C#

```
private System.Windows.Forms.ToolStripItem menuItemOrderByPathAscending;
```

#### Description

This is menuItemOrderByPathAscending, a member of class FrmMain.

### 1.1.2.3.2.14 FrmMain.menuItemOrderByPathDescending Field

C#

```
private System.Windows.Forms.ToolStripItem menuItemOrderByPathDescending;
```

#### Description

This is menuItemOrderByPathDescending, a member of class FrmMain.

### 1.1.2.3.2.15 FrmMain.menuStrip Field

C#

```
private System.Windows.Forms.ContextMenuStrip menuStrip;
```

**Description**

This is menuStrip, a member of class FrmMain.

**1.1.2.3.2.16 FrmMain.pbFiles Field****C#**

```
private System.Windows.Forms.ProgressBar pbFiles;
```

**Description**

This is pbFiles, a member of class FrmMain.

**1.1.2.3.2.17 FrmMain.resource Field****C#**

```
public static ResourceManager resource;
```

**Description**

This is resource, a member of class FrmMain.

**1.1.2.3.3 FrmMain Methods****1.1.2.3.3.1 FrmMain.btnAddFile\_Click Method**

It is used to add single or multiple PDF files to combine When you choose file or files The listbox in the form will be filled with the name of the files Which you choose to combine

**C#**

```
private void btnAddFile_Click(object sender, EventArgs e);
```

**Parameters**

| Parameters    | Description                             |
|---------------|---|
| object sender | The sender info (For example Main Form) |
| EventArgs e   | Event Arguments                         |

**Body Source**

```
1: private void btnAddFile_Click(object sender, EventArgs e)
2: {
3:     using (var dialogAddFile = new OpenFileDialog())
4:     {
5:         InitializeFileDialog(dialogAddFile);
6:         var result = dialogAddFile.ShowDialog();
7:         var addedFileCount = 0;
8:         if (result == DialogResult.OK && dialogAddFile.FileNames != null)
9:         {
10:            foreach (var file in dialogAddFile.FileNames)
11:            {
12:                if (!lbFiles.Items.Contains(file))
13:                {
14:                    lbFiles.Items.Add(file);
15:                    addedFileCount++;
16:                }
17:            }
18:            MessageBox.Show(addedFileCount + resource.GetString("FileAddMessage"),
19:                resource.GetString("AppTitle"), MessageBoxButtons.OK, MessageBoxIcon.Information);
20:        }
21:    }
```

### 1.1.2.3.3.2 FrmMain.btnAddFolder\_Click Method

This function is used to add PDF files in a folder which you choose in Folder Dialog recursively When you use it the added files will be seen in listbox

**C#**

```
private void btnAddFolder_Click(object sender, EventArgs e);
```

**Parameters**

| Parameters    | Description                             |
|---------------|---|
| object sender | The sender info (For example Main Form) |
| EventArgs e   | Event Arguments                         |

**Body Source**

```
1: private void btnAddFolder_Click(object sender, EventArgs e)
2: {
3:     using (var dialogAddFolder = new FolderBrowserDialog())
4:     {
5:         var result = dialogAddFolder.ShowDialog();
6:         if (result == DialogResult.OK &&
!string.IsNullOrEmpty(dialogAddFolder.SelectedPath))
7:         {
8:             var fileNames = Directory.GetFiles(dialogAddFolder.SelectedPath, "*.pdf",
SearchOption.AllDirectories);
9:             var addedFileCount = 0;
10:            foreach (var file in fileNames)
11:            {
12:                if (!lbFiles.Items.Contains(file))
13:                {
14:                    lbFiles.Items.Add(file);
15:                    addedFileCount++;
16:                }
17:            }
18:            MessageBox.Show(addedFileCount + resource.GetString("FileAddMessage"),
resource.GetString("AppTitle"), MessageBoxButtons.OK, MessageBoxIcon.Information);
19:        }
20:    }
21: }
```

### 1.1.2.3.3.3 FrmMain.btnClearList\_Click Method

This function is used to clear the file list in listbox

**C#**

```
private void btnClearList_Click(object sender, EventArgs e);
```

**Parameters**

| Parameters    | Description                             |
|---------------|---|
| object sender | The sender info (For example Main Form) |
| EventArgs e   | Event Arguments                         |

**Body Source**

```
1: private void btnClearList_Click(object sender, EventArgs e)
2: {
3:     var fileCount = lbFiles.Items.Count;
4:     lbFiles.Items.Clear();
5:     MessageBox.Show(fileCount + resource.GetString("DeleteFileMessage"),
resource.GetString("AppTitle"), MessageBoxButtons.OK, MessageBoxIcon.Information);
6: }
```

### 1.1.2.3.3.4 FrmMain.btnCombineITextSharp\_Click Method

This function is used to take list of PDF files in listbox, Combine them with a single file in location which you choose in folder dialog And you can see the progress in progress bar It uses iTextSharp Nuget Package

**C#**

```
private void btnCombineITextSharp_Click(object sender, EventArgs e);
```

**Parameters**

| Parameters    | Description                             |
|---------------|---|
| object sender | The sender info (For example Main Form) |
| EventArgs e   | Event Arguments                         |

**Body Source**

```
1: private void btnCombineITextSharp_Click(object sender, EventArgs e)
2: {
3:     try
4:     {
5:         if (lbFiles.Items.Count < 2)
6:             MessageBox.Show(resource.GetString("CombineWarning"),
resource.GetString("AppTitle"), MessageBoxButtons.OK, MessageBoxIcon.Error);
7:         else
8:         {
9:             using (var dialogExport = new FolderBrowserDialog())
10:            {
11:                var result = dialogExport.ShowDialog();
12:                if (result == DialogResult.OK &&
!string.IsNullOrEmpty(dialogExport.SelectedPath))
13:                {
14:                    var outputFileName = dialogExport.SelectedPath + "/" +
Guid.NewGuid() + ".pdf";
15:                    var fileCount = lbFiles.Items.Count;
16:                    var combinedFiles = 0;
17:
18:                    var outputFile = new Document();
19:                    using (FileStream outputFileStream = new FileStream(outputFileName,
FileMode.Create))
20:                    {
21:                        using (var pdfWriter = new
iTextSharp.text.pdf.PdfCopy(outputFile, outputFileStream))
22:                        {
23:                            if (pdfWriter == null)
24:                            {
25:                                return;
26:                            }
27:                            pdfWriter.SetFullCompression();
28:                            outputFile.Open();
29:                            for (int i = 0; i < lbFiles.Items.Count; i++)
30:                            {
31:                                try
32:                                {
33:                                    var pdfReader = new
iTextSharp.text.pdf.PdfReader(lbFiles.Items[i].ToString());
34:                                    pdfReader.ConsolidateNamedDestinations();
35:                                    for (int j = 1; j <= pdfReader.NumberOfPages; j++)
36:                                    {
37:                                        var page = pdfWriter.GetImportedPage(pdfReader,
j);
38:                                        pdfWriter.AddPage(page);
39:                                    }
40:                                    pdfReader.Close();
41:                                    combinedFiles++;
42:                                }
43:                                catch (Exception)
```

```

44:                {
45:                    fileCount--;
46:                }
47:
48:                pbFiles.Value = (combinedFiles * 100 / fileCount);
49:                if (pbFiles.Value > pbFiles.Maximum)
50:                    pbFiles.Value = pbFiles.Maximum;
51:                ActiveForm.Text = "%" + pbFiles.Value;
52:            }
53:            pdfWriter.Close();
54:        }
55:        outputFile.Close();
56:    }
57:
58:    MessageBox.Show(fileCount +
resource.GetString("CombineFileMessage") + outputFileName, resource.GetString("AppTitle"),
MessageBoxButtons.OK, MessageBoxIcon.Information);
59:
60:    pbFiles.Value = pbFiles.Minimum;
61:    ActiveForm.Text = resource.GetString("AppTitle");
62:    }
63:    }
64:    }
65:    }
66:    catch (Exception ex)
67:    {
68:        MessageBox.Show(resource.GetString("CombineErrorMessage") +
ex.GetAllMessages(), resource.GetString("AppTitle"), MessageBoxButtons.OK,
MessageBoxIcon.Error);
69:        pbFiles.Value = pbFiles.Minimum;
70:        ActiveForm.Text = resource.GetString("AppTitle");
71:    }
72: }

```

### 1.1.2.3.3.5 FrmMain.btnCombinePdfSharp\_Click Method

This function is used to take list of PDF files in listbox, Combine them with a single file in location which you choose in folder dialog And you can see the progress in progress bar It uses PdfSharp Nuget Package

**C#**

```
private void btnCombinePdfSharp_Click(object sender, EventArgs e);
```

#### Parameters

| Parameters    | Description                             |
|---------------|---|
| object sender | The sender info (For example Main Form) |
| EventArgs e   | Event Arguments                         |

#### Body Source

```

1: private void btnCombinePdfSharp_Click(object sender, EventArgs e)
2: {
3:     try
4:     {
5:         if (lbFiles.Items.Count < 2)
6:             MessageBox.Show(resource.GetString("CombineWarning"),
resource.GetString("AppTitle"), MessageBoxButtons.OK, MessageBoxIcon.Error);
7:         else
8:         {
9:             using (var dialogExport = new FolderBrowserDialog())
10:            {
11:                var result = dialogExport.ShowDialog();
12:                if (result == DialogResult.OK &&
!string.IsNullOrEmpty(dialogExport.SelectedPath))
13:                {
14:                    var outputFileName = dialogExport.SelectedPath + "/" +
Guid.NewGuid() + ".pdf";

```



```

15:         using (var outputFile = new PdfDocument())
16:         {
17:             outputFile.Options.CompressContentStreams = true;
18:             outputFile.Options.EnableCcittCompressionForBilevelImages =
true;
19:             outputFile.Options.FlateEncodeMode =
PdfFlateEncodeMode.BestCompression;
20:             var fileCount = lbFiles.Items.Count;
21:             var combinedFiles = 0;
22:
23:             for (int i = 0; i < lbFiles.Items.Count; i++)
24:             {
25:                 try
26:                 {
27:                     var inputDocument =
PdfReader.Open(lbFiles.Items[i].ToString(), PdfDocumentOpenMode.Import);
28:                     var count = inputDocument.PageCount;
29:                     for (int idx = 0; idx < count; idx++)
30:                     {
31:                         var page = inputDocument.Pages[idx];
32:                         outputFile.AddPage(page);
33:                     }
34:                     inputDocument.Close();
35:                     combinedFiles++;
36:                 }
37:                 catch (Exception)
38:                 {
39:                     fileCount--;
40:                 }
41:
42:                 pbFiles.Value = (combinedFiles * 100 / fileCount);
43:                 if (pbFiles.Value > pbFiles.Maximum)
44:                     pbFiles.Value = pbFiles.Maximum;
45:                 ActiveForm.Text = "%" + pbFiles.Value;
46:             }
47:             outputFile.Save(outputFileName);
48:             outputFile.Close();
49:
50:             MessageBox.Show(fileCount +
resource.GetString("CombineFileMessage") + outputFileName, resource.GetString("AppTitle"),
MessageBoxButtons.OK, MessageBoxIcon.Information);
51:         }
52:
53:         pbFiles.Value = pbFiles.Minimum;
54:         ActiveForm.Text = resource.GetString("AppTitle");
55:     }
56: }
57: }
58: }
59: catch (Exception ex)
60: {
61:     MessageBox.Show(resource.GetString("CombineErrorMessage") +
ex.GetAllMessages(), resource.GetString("AppTitle"), MessageBoxButtons.OK,
MessageBoxIcon.Error);
62:     pbFiles.Value = pbFiles.Minimum;
63:     ActiveForm.Text = resource.GetString("AppTitle");
64: }
65: }

```

### 1.1.2.3.3.6 FrmMain.cmbLanguage\_SelectedIndexChanged Method

This function is used to set the resource (see page 11) file And with this, you can see form elements, info messages etc. with the selected resource (see page 11) file contents. That is used to multi language support

**C#**

```
private void cmbLanguage_SelectedIndexChanged(object sender, EventArgs e);
```

**Parameters**

| Parameters    | Description                             |
|---------------|---|
| object sender | The sender info (For example Main Form) |
| EventArgs e   | Event Arguments                         |

**Body Source**

```

1: private void cmbLanguage_SelectedIndexChanged(object sender, EventArgs e)
2: {
3:     if (cmbLanguage.SelectedIndex != -1)
4:     {
5:         try
6:         {
7:             resource = new ResourceManager("PdfCombiner.Resources.AppResources-" +
cmbLanguage.SelectedItem.ToString().ToLower(), Assembly.GetExecutingAssembly());
8:             FormMembersNameInitialization();
9:         }
10:        catch (Exception)
11:        {
12:            resource = new ResourceManager("PdfCombiner.Resources.AppResources-en",
Assembly.GetExecutingAssembly());
13:            FormMembersNameInitialization();
14:        }
15:    }
16: }
17: }

```

**1.1.2.3.3.7 FrmMain.DeleteFilesFromListBox Method**

This method is used to Delete selected files from ListBox This is used different ways in this app So we created this as a method

**C#**

```
private void DeleteFilesFromListBox();
```

**Body Source**

```

1: private void DeleteFilesFromListBox()
2: {
3:     var allItems = new List<string>();
4:     var removedItems = new List<string>();
5:     var fileCount = lbFiles.SelectedItems.Count;
6:     for (int i = 0; i < lbFiles.SelectedItems.Count; i++)
7:     {
8:         removedItems.Add(lbFiles.SelectedItems[i].ToString());
9:     }
10:    foreach (var item in lbFiles.Items)
11:    {
12:        if (!removedItems.Any(j => j == item.ToString()))
13:            allItems.Add(item.ToString());
14:    }
15:    lbFiles.Items.Clear();
16:    foreach (var item in allItems)
17:    {
18:        lbFiles.Items.Add(item);
19:    }
20:    MessageBox.Show(fileCount + resource.GetString("DeleteFileMessage"),
resource.GetString("AppTitle"), MessageBoxButtons.OK, MessageBoxIcon.Information);
21: }

```

**1.1.2.3.3.8 FrmMain.Dispose Method**

Clean up any resources being used.

**C#**

```
protected override void Dispose(bool disposing);
```

**Parameters**

| Parameters     | Description   |
|----------------|---|
| bool disposing | true if managed resources should be disposed; otherwise, false. |

**Body Source**

```
1: protected override void Dispose(bool disposing)
2: {
3:     if (disposing && (components != null))
4:     {
5:         components.Dispose();
6:     }
7:     base.Dispose(disposing);
8: }
```

**1.1.2.3.3.9 FrmMain.FormMembersNameInitialization Method**

This method is used to change text of form members by the Resource file which is selected

**C#**

```
private void FormMembersNameInitialization();
```

**Body Source**

```
1: private void FormMembersNameInitialization()
2: {
3:     ActiveForm.Text = resource.GetString("AppTitle");
4:     btnAddFile.Text = resource.GetString("AddFile");
5:     btnAddFolder.Text = resource.GetString("AddFolder");
6:     btnClearList.Text = resource.GetString("ClearFileList");
7:     btnCombineITextSharp.Text = resource.GetString("CombineFilesITextSharp");
8:     btnCombinePdfSharp.Text = resource.GetString("CombineFilesPdfSharp");
9:     menuItemDelete.Text = resource.GetString("Delete");
10:    menuItemOrderByPathAscending.Text = resource.GetString("OrderByPathAscending");
11:    menuItemOrderByPathDescending.Text = resource.GetString("OrderByPathDescending");
12:    menuItemOrderByNameAscending.Text = resource.GetString("OrderByNameAscending");
13:    menuItemOrderByNameDescending.Text = resource.GetString("OrderByNameDescending");
14:    cmbLanguage.Text = resource.GetString("SelectLanguage");
15: }
```

**1.1.2.3.3.10 FrmMain.FrmMain\_FormClosed Method**

This function is used to terminate application

**C#**

```
private void FrmMain_FormClosed(object sender, FormClosedEventArgs e);
```

**Parameters**

| Parameters            | Description                             |
|-----------------------|---|
| object sender         | The sender info (For example Main Form) |
| FormClosedEventArgs e | Event Arguments                         |

**Body Source**

```
1: private void FrmMain_FormClosed(object sender, FormClosedEventArgs e)
2: {
3:     MessageBox.Show(resource.GetString("ThanksMessage"), resource.GetString("AppTitle"),
4:     MessageBoxButtons.OK, MessageBoxIcon.Information);
5:     Application.Exit();
6: }
```

### 1.1.2.3.3.11 FrmMain.FrmMain\_Shown Method

This function is used to assign menu to listbox When form is shown

**C#**

```
private void FrmMain_Shown(object sender, EventArgs e);
```

**Parameters**

| Parameters    | Description                             |
|---------------|---|
| object sender | The sender info (For example Main Form) |
| EventArgs e   | Event Arguments                         |

**Body Source**

```
1: private void FrmMain_Shown(object sender, EventArgs e)
2: {
3:     lbFiles.ContextMenuStrip = menuStrip;
4:     cmbLanguage.DropDownStyle = ComboBoxStyle.DropDownList;
5:     resource = new ResourceManager("PdfCombiner.Resources.AppResources-tr",
Assembly.GetExecutingAssembly());
6:     FormMembersNameInitialization();
7: }
```

### 1.1.2.3.3.12 FrmMain.InitializeComponent Method

Required method for Designer support - do not modify the contents of this method with the code editor.

**C#**

```
private void InitializeComponent();
```

**Body Source**

```
1: private void InitializeComponent()
2: {
3:     this.components = new System.ComponentModel.Container();
4:     System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(FrmMain));
5:     this.btnCombinePdfSharp = new System.Windows.Forms.Button();
6:     this.btnAddFolder = new System.Windows.Forms.Button();
7:     this.btnAddFile = new System.Windows.Forms.Button();
8:     this.btnClearList = new System.Windows.Forms.Button();
9:     this.lbFiles = new System.Windows.Forms.ListBox();
10:    this.lblDetails = new System.Windows.Forms.Label();
11:    this.pbFiles = new System.Windows.Forms.ProgressBar();
12:    this.btnCombineITextSharp = new System.Windows.Forms.Button();
13:    this.menuStrip = new System.Windows.Forms.ContextMenuStrip(this.components);
14:    this.menuItemDelete = new System.Windows.Forms.ToolStripMenuItem();
15:    this.menuItemOrderByPathAscending = new System.Windows.Forms.ToolStripMenuItem();
16:    this.menuItemOrderByPathDescending = new System.Windows.Forms.ToolStripMenuItem();
17:    this.menuItemOrderByNameAscending = new System.Windows.Forms.ToolStripMenuItem();
18:    this.menuItemOrderByNameDescending = new System.Windows.Forms.ToolStripMenuItem();
19:    this.cmbLanguage = new System.Windows.Forms.ComboBox();
20:    this.menuStrip.SuspendLayout();
21:    this.SuspendLayout();
22:    //
23:    // btnCombinePdfSharp
24:    //
25:    this.btnCombinePdfSharp.Image = ((System.Drawing.Image)
(resources.GetObject("btnCombinePdfSharp.Image")));
26:    this.btnCombinePdfSharp.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;
27:    this.btnCombinePdfSharp.Location = new System.Drawing.Point(442, 5);
28:    this.btnCombinePdfSharp.Name = "btnCombinePdfSharp";
29:    this.btnCombinePdfSharp.Size = new System.Drawing.Size(214, 37);
30:    this.btnCombinePdfSharp.TabIndex = 0;
31:    this.btnCombinePdfSharp.Text = "Combine Files (PdfSharp)";
32:    this.btnCombinePdfSharp.TextAlign = System.Drawing.ContentAlignment.MiddleRight;
```

```

33:         this.btnCombinePdfSharp.UseVisualStyleBackColor = true;
34:         this.btnCombinePdfSharp.Click += new
System.EventHandler(this.btnCombinePdfSharp_Click);
35:         //
36:         // btnAddFolder
37:         //
38:         this.btnAddFolder.Image = ((System.Drawing.Image)
(resources.GetObject("btnAddFolder.Image")));
39:         this.btnAddFolder.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;
40:         this.btnAddFolder.Location = new System.Drawing.Point(197, 5);
41:         this.btnAddFolder.Name = "btnAddFolder";
42:         this.btnAddFolder.Size = new System.Drawing.Size(171, 37);
43:         this.btnAddFolder.TabIndex = 2;
44:         this.btnAddFolder.Text = "Add Folder";
45:         this.btnAddFolder.TextAlign = System.Drawing.ContentAlignment.MiddleRight;
46:         this.btnAddFolder.UseVisualStyleBackColor = true;
47:         this.btnAddFolder.Click += new System.EventHandler(this.btnAddFolder_Click);
48:         //
49:         // btnAddFile
50:         //
51:         this.btnAddFile.Image = ((System.Drawing.Image)
(resources.GetObject("btnAddFile.Image")));
52:         this.btnAddFile.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;
53:         this.btnAddFile.Location = new System.Drawing.Point(12, 5);
54:         this.btnAddFile.Name = "btnAddFile";
55:         this.btnAddFile.Size = new System.Drawing.Size(179, 37);
56:         this.btnAddFile.TabIndex = 3;
57:         this.btnAddFile.Text = "Add Files";
58:         this.btnAddFile.TextAlign = System.Drawing.ContentAlignment.MiddleRight;
59:         this.btnAddFile.UseVisualStyleBackColor = true;
60:         this.btnAddFile.Click += new System.EventHandler(this.btnAddFile_Click);
61:         //
62:         // btnClearList
63:         //
64:         this.btnClearList.Image = ((System.Drawing.Image)
(resources.GetObject("btnClearList.Image")));
65:         this.btnClearList.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;
66:         this.btnClearList.Location = new System.Drawing.Point(12, 45);
67:         this.btnClearList.Name = "btnClearList";
68:         this.btnClearList.Size = new System.Drawing.Size(179, 37);
69:         this.btnClearList.TabIndex = 4;
70:         this.btnClearList.Text = "Clear File List";
71:         this.btnClearList.TextAlign = System.Drawing.ContentAlignment.MiddleRight;
72:         this.btnClearList.UseVisualStyleBackColor = true;
73:         this.btnClearList.Click += new System.EventHandler(this.btnClearList_Click);
74:         //
75:         // lbFiles
76:         //
77:         this.lbFiles.AllowDrop = true;
78:         this.lbFiles.FormattingEnabled = true;
79:         this.lbFiles.Location = new System.Drawing.Point(12, 88);
80:         this.lbFiles.Name = "lbFiles";
81:         this.lbFiles.SelectionMode = System.Windows.Forms.SelectionMode.MultiExtended;
82:         this.lbFiles.Size = new System.Drawing.Size(644, 186);
83:         this.lbFiles.TabIndex = 5;
84:         this.lbFiles.DragDrop += new
System.Windows.Forms.DragEventHandler(this.lbFiles_DragDrop);
85:         this.lbFiles.DragOver += new
System.Windows.Forms.DragEventHandler(this.lbFiles_DragOver);
86:         this.lbFiles.KeyDown += new
System.Windows.Forms.KeyEventHandler(this.lbFiles_KeyDown);
87:         this.lbFiles.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.lbFiles_MouseDown);
88:         //
89:         // lblDetails
90:         //
91:         this.lblDetails.AutoSize = true;
92:         this.lblDetails.Location = new System.Drawing.Point(12, 65);
93:         this.lblDetails.Name = "lblDetails";

```

```

94:         this.lblDetails.Size = new System.Drawing.Size(0, 13);
95:         this.lblDetails.TabIndex = 6;
96:         //
97:         // pbFiles
98:         //
99:         this.pbFiles.Location = new System.Drawing.Point(197, 45);
100:        this.pbFiles.Name = "pbFiles";
101:        this.pbFiles.Size = new System.Drawing.Size(239, 37);
102:        this.pbFiles.TabIndex = 7;
103:        //
104:        // btnCombineITextSharp
105:        //
106:        this.btnCombineITextSharp.Image = ((System.Drawing.Image)
(resources.GetObject("btnCombineITextSharp.Image")));
107:        this.btnCombineITextSharp.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;
108:        this.btnCombineITextSharp.Location = new System.Drawing.Point(442, 45);
109:        this.btnCombineITextSharp.Name = "btnCombineITextSharp";
110:        this.btnCombineITextSharp.Size = new System.Drawing.Size(214, 37);
111:        this.btnCombineITextSharp.TabIndex = 9;
112:        this.btnCombineITextSharp.Text = "Combine Files (iTextSharp)";
113:        this.btnCombineITextSharp.TextAlign = System.Drawing.ContentAlignment.MiddleRight;
114:        this.btnCombineITextSharp.UseVisualStyleBackColor = true;
115:        this.btnCombineITextSharp.Click += new
System.EventHandler(this.btnCombineITextSharp_Click);
116:        //
117:        // menuStrip
118:        //
119:        this.menuStrip.Items.AddRange(new System.Windows.Forms.ToolStripItem[]
{this.menuItemDelete, this.menuItemOrderByPathAscending,
this.menuItemOrderByPathDescending, this.menuItemOrderByNameAscending,
this.menuItemOrderByNameDescending});
120:        this.menuStrip.Name = "menuStrip";
121:        this.menuStrip.Size = new System.Drawing.Size(229, 114);
122:        //
123:        // menuItemDelete
124:        //
125:        this.menuItemDelete.Image = ((System.Drawing.Image)
(resources.GetObject("menuItemDelete.Image")));
126:        this.menuItemDelete.Name = "menuItemDelete";
127:        this.menuItemDelete.Size = new System.Drawing.Size(228, 22);
128:        this.menuItemDelete.Text = "Delete";
129:        this.menuItemDelete.Click += new System.EventHandler(this.menuItemDelete_Click);
130:        //
131:        // menuItemOrderByPathAscending
132:        //
133:        this.menuItemOrderByPathAscending.Image = ((System.Drawing.Image)
(resources.GetObject("menuItemOrderByPathAscending.Image")));
134:        this.menuItemOrderByPathAscending.Name = "menuItemOrderByPathAscending";
135:        this.menuItemOrderByPathAscending.Size = new System.Drawing.Size(228, 22);
136:        this.menuItemOrderByPathAscending.Text = "Order By Path (Ascending)";
137:        this.menuItemOrderByPathAscending.Click += new
System.EventHandler(this.menuItemOrderByPathAscending_Click);
138:        //
139:        // menuItemOrderByPathDescending
140:        //
141:        this.menuItemOrderByPathDescending.Image = ((System.Drawing.Image)
(resources.GetObject("menuItemOrderByPathDescending.Image")));
142:        this.menuItemOrderByPathDescending.Name = "menuItemOrderByPathDescending";
143:        this.menuItemOrderByPathDescending.Size = new System.Drawing.Size(228, 22);
144:        this.menuItemOrderByPathDescending.Text = "Order By Path (Descending)";
145:        this.menuItemOrderByPathDescending.Click += new
System.EventHandler(this.menuItemOrderByPathDescending_Click);
146:        //
147:        // menuItemOrderByNameAscending
148:        //
149:        this.menuItemOrderByNameAscending.Image = ((System.Drawing.Image)
(resources.GetObject("menuItemOrderByNameAscending.Image")));
150:        this.menuItemOrderByNameAscending.Name = "menuItemOrderByNameAscending";
151:        this.menuItemOrderByNameAscending.Size = new System.Drawing.Size(228, 22);

```



```

152:         this.menuItemOrderByNameAscending.Text = "Order By Name (Ascending)";
153:         this.menuItemOrderByNameAscending.Click += new
System.EventHandler(this.menuItemOrderByNameAscending_Click);
154:         //
155:         // menuItemOrderByNameDescending
156:         //
157:         this.menuItemOrderByNameDescending.Image = ((System.Drawing.Image)
(resources.GetObject("menuItemOrderByNameDescending.Image")));
158:         this.menuItemOrderByNameDescending.Name = "menuItemOrderByNameDescending";
159:         this.menuItemOrderByNameDescending.Size = new System.Drawing.Size(228, 22);
160:         this.menuItemOrderByNameDescending.Text = "Order By Name (Descending)";
161:         this.menuItemOrderByNameDescending.Click += new
System.EventHandler(this.menuItemOrderByNameDescending_Click);
162:         //
163:         // cmbLanguage
164:         //
165:         this.cmbLanguage.Anchor = System.Windows.Forms.AnchorStyles.None;
166:         this.cmbLanguage.FormattingEnabled = true;
167:         this.cmbLanguage.Items.AddRange(new object[] { "TR", "EN", "DE", "FR", "RU", "ES" });
168:         this.cmbLanguage.Location = new System.Drawing.Point(374, 12);
169:         this.cmbLanguage.Name = "cmbLanguage";
170:         this.cmbLanguage.Size = new System.Drawing.Size(62, 21);
171:         this.cmbLanguage.TabIndex = 10;
172:         this.cmbLanguage.Text = "Select Language";
173:         this.cmbLanguage.SelectedIndexChanged += new
System.EventHandler(this.cmbLanguage_SelectedIndexChanged);
174:         //
175:         // FrmMain
176:         //
177:         this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
178:         this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
179:         this.ClientSize = new System.Drawing.Size(668, 287);
180:         this.Controls.Add(this.cmbLanguage);
181:         this.Controls.Add(this.btnCombineITextSharp);
182:         this.Controls.Add(this.pbFiles);
183:         this.Controls.Add(this.lblDetails);
184:         this.Controls.Add(this.lbFiles);
185:         this.Controls.Add(this.btnClearList);
186:         this.Controls.Add(this.btnAddFile);
187:         this.Controls.Add(this.btnAddFolder);
188:         this.Controls.Add(this.btnCombinePdfSharp);
189:         this.Name = "FrmMain";
190:         this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
191:         this.Text = "PDF Combiner";
192:         this.FormClosed += new
System.Windows.Forms.FormClosedEventHandler(this.FrmMain_FormClosed);
193:         this.Shown += new System.EventHandler(this.FrmMain_Shown);
194:         this.menuStrip.ResumeLayout(false);
195:         this.ResumeLayout(false);
196:         this.PerformLayout();
197: }

```

### 1.1.2.3.3.13 FrmMain.InitializeFileDialog Method

This function is used to set options of file dialog like Filter, start path etc.

**C#**

```
private static void InitializeFileDialog(OpenFileDialog dialogAddFile);
```

**Parameters**

| Parameters                   | Description |
|------------------------------|-------------|
| OpenFileDialog dialogAddFile | File Dialog |

**Body Source**

```

1: private static void InitializeFileDialog(OpenFileDialog dialogAddFile)
2: {

```

```

3:     dialogAddFile.Multiselect = true;
4:     dialogAddFile.Filter = resource.GetString("PdfFiles") + " (*.pdf)|*.pdf";
5:     dialogAddFile.InitialDirectory = Application.StartupPath;
6:     dialogAddFile.Title = resource.GetString("SelectPdfFile");
7:     dialogAddFile.DefaultExt = "PDF";
8: }

```

### 1.1.2.3.3.14 FrmMain.lbFiles\_DragDrop Method

This is used to drag and drop an item in listbox Which is used to order books and combine them with that order

**C#**

```
private void lbFiles_DragDrop(object sender, DragEventArgs e);
```

**Parameters**

| Parameters      | Description                             |
|-----------------|---|
| object sender   | The sender info (For example Main Form) |
| DragEventArgs e | Event Arguments                         |

**Description**

m>

**Body Source**

```

1: private void lbFiles_DragDrop(object sender, DragEventArgs e)
2: {
3:     var point = lbFiles.PointToClient(new Point(e.X, e.Y));
4:     var index = this.lbFiles.IndexFromPoint(point);
5:     if (index < 0) index = this.lbFiles.Items.Count - 1;
6:     // type of string because file names stored in string in listbox
7:     var data = e.Data.GetData(typeof(string));
8:     this.lbFiles.Items.Remove(data);
9:     this.lbFiles.Items.Insert(index, data);
10: }

```

### 1.1.2.3.3.15 FrmMain.lbFiles\_DragOver Method

This is used to give effect while dragging and dropping an item in listbox

**C#**

```
private void lbFiles_DragOver(object sender, DragEventArgs e);
```

**Parameters**

| Parameters      | Description                             |
|-----------------|---|
| object sender   | The sender info (For example Main Form) |
| DragEventArgs e | Event Arguments                         |

**Body Source**

```

1: private void lbFiles_DragOver(object sender, DragEventArgs e)
2: {
3:     e.Effect = DragDropEffects.Move;
4: }

```

### 1.1.2.3.3.16 FrmMain.lbFiles\_KeyDown Method

This function is used to delete files in listbox which are chosen If you press the Delete button in your keyboard The selected files will be deleted from the list

**C#**

```
private void lbFiles_KeyDown(object sender, KeyEventArgs e);
```



**Parameters**

| Parameters     | Description                             |
|----------------|---|
| object sender  | The sender info (For example Main Form) |
| KeyEventArgs e | Event Arguments                         |

**Body Source**

```
1: private void lbFiles_KeyDown(object sender, KeyEventArgs e)
2: {
3:     if (e.KeyValue == Keys.Delete.GetHashCode())
4:         DeleteFilesFromListBox();
5: }
```

**1.1.2.3.3.17 FrmMain.lbFiles\_MouseDown Method**

This is used to control mouse actions and if that is not right click Start to drag and drop item in listbox

**C#**

```
private void lbFiles_MouseDown(object sender, MouseEventArgs e);
```

**Parameters**

| Parameters       | Description                             |
|------------------|---|
| object sender    | The sender info (For example Main Form) |
| MouseEventArgs e | Event Arguments                         |

**Body Source**

```
1: private void lbFiles_MouseDown(object sender, MouseEventArgs e)
2: {
3:     if (!e.Button.Equals(MouseButtons.Right))
4:     {
5:         if (lbFiles.SelectedItem == null) return;
6:         lbFiles.DoDragDrop(lbFiles.SelectedItem, DragDropEffects.Move);
7:     }
8: }
```

**1.1.2.3.3.18 FrmMain.menuItemDelete\_Click Method**

This is used to delete selected items in listbox When menu item Delete is clicked and opened dialog

**C#**

```
private void menuItemDelete_Click(object sender, EventArgs e);
```

**Parameters**

| Parameters    | Description                             |
|---------------|---|
| object sender | The sender info (For example Main Form) |
| EventArgs e   | Event Arguments                         |

**Body Source**

```
1: private void menuItemDelete_Click(object sender, EventArgs e)
2: {
3:     if (lbFiles.Items.Count > 0)
4:     {
5:         if (lbFiles.SelectedItems.Count > 0)
6:         {
7:             var deleteDialog = new DialogResult();
8:             deleteDialog = MessageBox.Show(resource.GetString("DeleteWarning1") +
lbFiles.SelectedItems.Count + resource.GetString("DeleteWarning2"),
resource.GetString("AppTitle"), MessageBoxButtons.YesNo, MessageBoxIcon.Warning);
9:             if (deleteDialog == DialogResult.Yes)
10:                 DeleteFilesFromListBox();
11:         }
12: }
```

```
12:         else
13:             MessageBox.Show(resource.GetString("NoSelectedFile"),
resource.GetString("AppTitle"), MessageBoxButtons.OK, MessageBoxIcon.Error);
14:     }
15:     else
16:         MessageBox.Show(resource.GetString("NoFile"), resource.GetString("AppTitle"),
MessageBoxButtons.OK, MessageBoxIcon.Error);
17: }
```

### 1.1.2.3.3.19 FrmMain.menuItemOrderByNameAscending\_Click Method

This method is used to order elements in listbox ascending by file name

**C#**

```
private void menuItemOrderByNameAscending_Click(object sender, EventArgs e);
```

**Parameters**

| Parameters    | Description                             |
|---------------|---|
| object sender | The sender info (For example Main Form) |
| EventArgs e   | Event Arguments                         |

**Body Source**

```
1: private void menuItemOrderByNameAscending_Click(object sender, EventArgs e)
2: {
3:     if (lbFiles.Items.Count > 0)
4:         lbFiles = SortItemsByName(lbFiles, true);
5:     else
6:         MessageBox.Show(resource.GetString("NoFile"), resource.GetString("AppTitle"),
MessageBoxButtons.OK, MessageBoxIcon.Error);
7: }
```

### 1.1.2.3.3.20 FrmMain.menuItemOrderByNameDescending\_Click Method

This method is used to order elements in listbox descending by file name

**C#**

```
private void menuItemOrderByNameDescending_Click(object sender, EventArgs e);
```

**Parameters**

| Parameters    | Description                             |
|---------------|---|
| object sender | The sender info (For example Main Form) |
| EventArgs e   | Event Arguments                         |

**Body Source**

```
1: private void menuItemOrderByNameDescending_Click(object sender, EventArgs e)
2: {
3:     if (lbFiles.Items.Count > 0)
4:         lbFiles = SortItemsByName(lbFiles, false);
5:     else
6:         MessageBox.Show(resource.GetString("NoFile"), resource.GetString("AppTitle"),
MessageBoxButtons.OK, MessageBoxIcon.Error);
7: }
```

### 1.1.2.3.3.21 FrmMain.menuItemOrderByPathAscending\_Click Method

This method is used to order elements in listbox ascending by full path

**C#**

```
private void menuItemOrderByPathAscending_Click(object sender, EventArgs e);
```

**Parameters**

| Parameters    | Description                             |
|---------------|---|
| object sender | The sender info (For example Main Form) |
| EventArgs e   | Event Arguments                         |

**Body Source**

```

1: private void menuItemOrderByPathAscending_Click(object sender, EventArgs e)
2: {
3:     if (lbFiles.Items.Count > 0)
4:         lbFiles = SortItemsByPath(lbFiles, true);
5:     else
6:         MessageBox.Show(resource.GetString("NoFile"), resource.GetString("AppTitle"),
7:             MessageBoxButtons.OK, MessageBoxIcon.Error);
8: }

```

**1.1.2.3.3.22 FrmMain.menuItemOrderByPathDescending\_Click Method**

This method is used to order elements in listbox descending by full path

**C#**

```
private void menuItemOrderByPathDescending_Click(object sender, EventArgs e);
```

**Parameters**

| Parameters    | Description                             |
|---------------|---|
| object sender | The sender info (For example Main Form) |
| EventArgs e   | Event Arguments                         |

**Body Source**

```

1: private void menuItemOrderByPathDescending_Click(object sender, EventArgs e)
2: {
3:     if (lbFiles.Items.Count > 0)
4:         lbFiles = SortItemsByPath(lbFiles, false);
5:     else
6:         MessageBox.Show(resource.GetString("NoFile"), resource.GetString("AppTitle"),
7:             MessageBoxButtons.OK, MessageBoxIcon.Error);
8: }

```

**1.1.2.3.3.23 FrmMain.SortItemsByName Method**

This method is used to order items by file name in listbox And return them with the given order as parametre

**C#**

```
private static ListBox SortItemsByName(ListBox listBox, bool ascending);
```

**Parameters**

| Parameters      | Description                    |
|-----------------|--------------------------------|
| Listbox listBox | Listbox Info                   |
| bool ascending  | Order Type is Ascending Or Not |

**Body Source**

```

1: private static ListBox SortItemsByName(ListBox listBox, bool ascending)
2: {
3:     var fileNameList = new List<string>();
4:     List<object> items;
5:     items = listBox.Items.OfType<object>().ToList();
6:     const char c = '\u005c';
7:     foreach (var item in items)
8:     {
9:         var fullPath = item.ToString();
10:        var list = fullPath.Split(c).ToList();

```

```

11:         if (list.Count > 0)
12:             fileNameList.Add(list[list.Count - 1]);
13:     }
14:
15:     if (ascending)
16:         fileNameList = fileNameList.OrderBy(j => j).ToList();
17:     else
18:         fileNameList = fileNameList.OrderByDescending(j => j).ToList();
19:
20:     listBox.Items.Clear();
21:     foreach (var item in fileNameList)
22:         listBox.Items.Add(items.First(j => j.ToString().EndsWith(item)));
23:
24:     return listBox;
25: }

```

### 1.1.2.3.3.24 FrmMain.SortItemsByPath Method

This method is used to order items by full path in listbox And return them with the given order as parametre

**C#**

```
private static ListBox SortItemsByPath(ListBox listBox, bool ascending);
```

**Parameters**

| Parameters      | Description                    |
|-----------------|--------------------------------|
| ListBox listBox | ListBox Info                   |
| bool ascending  | Order Type is Ascending Or Not |

**Body Source**

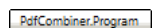
```

1: private static ListBox SortItemsByPath(ListBox listBox, bool ascending)
2: {
3:     List<object> items;
4:     items = listBox.Items.OfType<object>().ToList();
5:     listBox.Items.Clear();
6:     if (ascending)
7:         listBox.Items.AddRange(items.OrderBy(i => i).ToArray());
8:     else
9:         listBox.Items.AddRange(items.OrderByDescending(i => i).ToArray());
10:    return listBox;
11: }

```

## 1.1.2.4 Program Class

**Class Hierarchy**

 PdfCombiner Program

**C#**

```
public static class Program;
```

**File**

Program.cs ( see page 43)

**Description**

This is class PdfCombiner.Program.

**Program Methods**

|   | Name  | Description                               |
|---|---|---|
|  | Main (  see page 27) | The main entry point for the application. |

### 1.1.2.4.1 Program Methods

#### 1.1.2.4.1.1 Program.Main Method

The main entry point for the application.

**C#**

```
[STAThread]
private static void Main();
```

**Body Source**

```
1: [STAThread]
2: static void Main()
3: {
4:     Application.EnableVisualStyles();
5:     Application.SetCompatibleTextRenderingDefault(false);
6:     Application.Run(new FrmMain());
7: }
```

## 1.2 Files

The following table lists files in this documentation.

**Files**

| Name                                 | Description  |
|--------------------------------------|--|
| AssemblyInfo.cs (see page 27)        | This is file AssemblyInfo.cs.  |
| BaseException.cs (see page 28)       | This is file BaseException.cs.   |
| ExceptionExtensions.cs (see page 29) | This is file ExceptionExtensions.cs.   |
| FrmMain.cs (see page 29)             | This is file FrmMain.cs.   |
| FrmMain.Designer.cs (see page 38)    | This is file FrmMain.Designer.cs.  |
| PdfCombiner.csproj (see page 43)     | This is file PdfCombiner.csproj.   |
| PdfCombiner.sln (see page 43)        | This is file PdfCombiner.sln.  |
| Program.cs (see page 43)             | This is file Program.cs.   |
| Resources.Designer.cs (see page 44)  | This code was generated by a tool. Runtime Version:4.0.30319.42000<br>Changes to this file may cause incorrect behavior and will be lost if the code is regenerated. |
| Settings.Designer.cs (see page 45)   | This code was generated by a tool. Runtime Version:4.0.30319.42000<br>Changes to this file may cause incorrect behavior and will be lost if the code is regenerated. |

### 1.2.1 AssemblyInfo.cs

This is file AssemblyInfo.cs.

**Body Source**

```
1: ?using System.Reflection;
2: using System.Runtime.CompilerServices;
3: using System.Runtime.InteropServices;
4:
```

```

5: // General Information about an assembly is controlled through the following
6: // set of attributes. Change these attribute values to modify the information
7: // associated with an assembly.
8: [assembly: AssemblyTitle("PdfCombiner")]
9: [assembly: AssemblyDescription("")]
10: [assembly: AssemblyConfiguration("")]
11: [assembly: AssemblyCompany("")]
12: [assembly: AssemblyProduct("PdfCombiner")]
13: [assembly: AssemblyCopyright("Copyright © 2022")]
14: [assembly: AssemblyTrademark("")]
15: [assembly: AssemblyCulture("")]
16:
17: // Setting ComVisible to false makes the types in this assembly not visible
18: // to COM components. If you need to access a type in this assembly from
19: // COM, set the ComVisible attribute to true on that type.
20: [assembly: ComVisible(false)]
21:
22: // The following GUID is for the ID of the typelib if this project is exposed to COM
23: [assembly: Guid("48e34ee8-dfaa-48db-acfd-70573cee9c00")]
24:
25: // Version information for an assembly consists of the following four values:
26: //
27: //      Major Version
28: //      Minor Version
29: //      Build Number
30: //      Revision
31: //
32: // You can specify all the values or you can default the Build and Revision Numbers
33: // by using the '*' as shown below:
34: // [assembly: AssemblyVersion("1.0.*")]
35: [assembly: AssemblyVersion("1.0.0.0")]
36: [assembly: AssemblyFileVersion("1.0.0.0")]

```

## 1.2.2 BaseException.cs

This is file BaseException.cs.

### Body Source

```

1: ?using System;
2:
3: namespace PdfCombiner
4: {
5:     /// <summary>
6:     /// This function is used to take exceptions which can be thrown in
7:     /// PDF Combiner App
8:     /// </summary>
9:     [Serializable]
10:    public abstract class BaseException : Exception
11:    {
12:        protected BaseException()
13:        {
14:        }
15:
16:        protected BaseException(string message) : base(message)
17:        {
18:        }
19:
20:        protected BaseException(string message, Exception innerException) :
21:        base(message, innerException)
22:        {
23:        }
24:    }

```

**Namespaces**

| Name                     | Description                    |
|--------------------------|--------------------------------|
| PdfCombiner (see page 1) | This is namespace PdfCombiner. |

## 1.2.3 ExceptionExtensions.cs

This is file ExceptionExtensions.cs.

**Body Source**

```
1: ?using System;
2:
3: namespace PdfCombiner
4: {
5:     public static class ExceptionExtensions
6:     {
7:         /// <summary>
8:         /// This method is used to take all messages with stack trace info in exception
9:         /// </summary>
10:        /// <param name="exception">Exception</param>
11:        /// <returns>Detailed Error Message</returns>
12:        public static string GetAllMessages(this Exception exception)
13:        {
14:            var messages = exception.Message;
15:            if (exception.InnerException != null)
16:                messages += " " + exception.InnerException.GetAllMessages();
17:            if (!string.IsNullOrEmpty(exception.StackTrace))
18:                messages += " " + exception.StackTrace;
19:            if (!string.IsNullOrEmpty(exception.Source))
20:                messages += " " + exception.Source;
21:            return messages;
22:        }
23:    }
24: }
```

**Namespaces**

| Name                     | Description                    |
|--------------------------|--------------------------------|
| PdfCombiner (see page 1) | This is namespace PdfCombiner. |

## 1.2.4 FrmMain.cs

This is file FrmMain.cs.

**Body Source**

```
1: ?using iTextSharp.text;
2: using PdfSharp.Pdf;
3: using PdfSharp.Pdf.IO;
4: using System;
5: using System.Collections.Generic;
6: using System.Drawing;
7: using System.IO;
8: using System.Linq;
9: using System.Reflection;
10: using System.Resources;
11: using System.Windows.Forms;
12:
13: namespace PdfCombiner
```

```

14: {
15:     public partial class FrmMain : Form
16:     {
17:         public static ResourceManager resource;
18:
19:         public FrmMain()
20:         {
21:             InitializeComponent();
22:         }
23:
24:         #region Form Init, Language Settings and Finalize Methods
25:
26:         /// <summary>
27:         /// This function is used to terminate application
28:         /// </summary>
29:         /// <param name="sender">The sender info (For example Main Form)</param>
30:         /// <param name="e">Event Arguments</param>
31:         private void FrmMain_FormClosed(object sender, FormClosedEventArgs e)
32:         {
33:             MessageBox.Show(resource.GetString("ThanksMessage"),
resource.GetString("AppTitle"), MessageBoxButtons.OK, MessageBoxIcon.Information);
34:             Application.Exit();
35:         }
36:
37:         /// <summary>
38:         /// This function is used to assign menu to listbox
39:         /// When form is shown
40:         /// </summary>
41:         /// <param name="sender">The sender info (For example Main Form)</param>
42:         /// <param name="e">Event Arguments</param>
43:         private void FrmMain_Shown(object sender, EventArgs e)
44:         {
45:             lbFiles.ContextMenuStrip = menuStrip;
46:             cmbLanguage.DropDownStyle = ComboBoxStyle.DropDownList;
47:             resource = new ResourceManager("PdfCombiner.Resources.AppResources-tr",
Assembly.GetExecutingAssembly());
48:             FormMembersNameInitialization();
49:         }
50:
51:         /// <summary>
52:         /// This function is used to set the resource file
53:         /// And with this, you can see form elements, info messages etc. with the
selected
54:         /// resource file contents. That is used to multi language support
55:         /// </summary>
56:         /// <param name="sender">The sender info (For example Main Form)</param>
57:         /// <param name="e">Event Arguments</param>
58:         private void cmbLanguage_SelectedIndexChanged(object sender, EventArgs e)
59:         {
60:             if (cmbLanguage.SelectedIndex != -1)
61:             {
62:                 try
63:                 {
64:                     resource = new
ResourceManager("PdfCombiner.Resources.AppResources-" +
cmbLanguage.SelectedItem.ToString().ToLower(), Assembly.GetExecutingAssembly());
65:                     FormMembersNameInitialization();
66:                 }
67:                 catch (Exception)
68:                 {
69:                     resource = new
ResourceManager("PdfCombiner.Resources.AppResources-en", Assembly.GetExecutingAssembly());
70:                     FormMembersNameInitialization();
71:                 }
72:             }
73:         }
74:     }
75:
76:     /// <summary>

```



```

77:         /// This method is used to change text of form members by the
78:         /// Resource file which is selected
79:         /// </summary>
80:         private void FormMembersNameInitialization()
81:         {
82:             ActiveForm.Text = resource.GetString("AppTitle");
83:             btnAddFile.Text = resource.GetString("AddFile");
84:             btnAddFolder.Text = resource.GetString("AddFolder");
85:             btnClearList.Text = resource.GetString("ClearFileList");
86:             btnCombineITextSharp.Text = resource.GetString("CombineFilesITextSharp");
87:             btnCombinePdfSharp.Text = resource.GetString("CombineFilesPdfSharp");
88:             menuItemDelete.Text = resource.GetString("Delete");
89:             menuItemOrderByPathAscending.Text =
resource.GetString("OrderByPathAscending");
90:             menuItemOrderByPathDescending.Text =
resource.GetString("OrderByPathDescending");
91:             menuItemOrderByNameAscending.Text =
resource.GetString("OrderByNameAscending");
92:             menuItemOrderByNameDescending.Text =
resource.GetString("OrderByNameDescending");
93:             cmbLanguage.Text = resource.GetString("SelectLanguage");
94:         }
95:
96:         #endregion
97:
98:         #region Add Item Methods
99:
100:        /// <summary>
101:        /// It is used to add single or multiple PDF files to combine
102:        /// When you choose file or files
103:        /// The listbox in the form will be filled with the name of the files
104:        /// Which you choose to combine
105:        /// </summary>
106:        /// <param name="sender">The sender info (For example Main Form)</param>
107:        /// <param name="e">Event Arguments</param>
108:        private void btnAddFile_Click(object sender, EventArgs e)
109:        {
110:            using (var dialogAddFile = new OpenFileDialog())
111:            {
112:                InitializeFileDialog(dialogAddFile);
113:                var result = dialogAddFile.ShowDialog();
114:                var addedFileCount = 0;
115:                if (result == DialogResult.OK && dialogAddFile.FileNames != null)
116:                {
117:                    foreach (var file in dialogAddFile.FileNames)
118:                    {
119:                        if (!lbFiles.Items.Contains(file))
120:                        {
121:                            lbFiles.Items.Add(file);
122:                            addedFileCount++;
123:                        }
124:                    }
125:                    MessageBox.Show(addedFileCount +
resource.GetString("FileAddMessage"), resource.GetString("AppTitle"), MessageBoxButtons.OK,
MessageBoxIcon.Information);
126:                }
127:            }
128:        }
129:
130:        /// <summary>
131:        /// This function is used to set options of file dialog like
132:        /// Filter, start path etc.
133:        /// </summary>
134:        /// <param name="dialogAddFile">File Dialog</param>
135:        private static void InitializeFileDialog(OpenFileDialog dialogAddFile)
136:        {
137:            dialogAddFile.Multiselect = true;
138:            dialogAddFile.Filter = resource.GetString("PdfFiles") + " (*.pdf)|*.pdf";
139:            dialogAddFile.InitialDirectory = Application.StartupPath;

```

```

140:         dialogAddFile.Title = resource.GetString("SelectPdfFile");
141:         dialogAddFile.DefaultExt = "PDF";
142:     }
143:
144:     /// <summary>
145:     /// This function is used to add PDF files in a folder which you choose in
146:     /// Folder Dialog recursively
147:     /// When you use it the added files will be seen in listbox
148:     /// </summary>
149:     /// <param name="sender">The sender info (For example Main Form)</param>
150:     /// <param name="e">Event Arguments</param>
151:     private void btnAddFolder_Click(object sender, EventArgs e)
152:     {
153:         using (var dialogAddFolder = new FolderBrowserDialog())
154:         {
155:             var result = dialogAddFolder.ShowDialog();
156:             if (result == DialogResult.OK &&
!string.IsNullOrEmpty(dialogAddFolder.SelectedPath))
157:             {
158:                 var fileNames = Directory.GetFiles(dialogAddFolder.SelectedPath,
159:                 "*.pdf", SearchOption.AllDirectories);
160:                 var addedFileCount = 0;
161:                 foreach (var file in fileNames)
162:                 {
163:                     if (!lbFiles.Items.Contains(file))
164:                     {
165:                         lbFiles.Items.Add(file);
166:                         addedFileCount++;
167:                     }
168:                 }
169:                 MessageBox.Show(addedFileCount +
170:                 resource.GetString("FileAddMessage"), resource.GetString("AppTitle"), MessageBoxButtons.OK,
171:                 MessageBoxIcon.Information);
172:             }
173:         }
174:     }
175:     #endregion
176:     #region Combine PDF Methods
177:
178:     /// <summary>
179:     /// This function is used to take list of PDF files in listbox,
180:     /// Combine them with a single file in location which you choose in folder
181:     /// dialog
182:     /// And you can see the progress in progress bar
183:     /// It uses PdfSharp Nuget Package
184:     /// </summary>
185:     /// <param name="sender">The sender info (For example Main Form)</param>
186:     /// <param name="e">Event Arguments</param>
187:     private void btnCombinePdfSharp_Click(object sender, EventArgs e)
188:     {
189:         try
190:         {
191:             if (lbFiles.Items.Count < 2)
192:                 MessageBox.Show(resource.GetString("CombineWarning"),
193:                 resource.GetString("AppTitle"), MessageBoxButtons.OK, MessageBoxIcon.Error);
194:             else
195:             {
196:                 using (var dialogExport = new FolderBrowserDialog())
197:                 {
198:                     var result = dialogExport.ShowDialog();
199:                     if (result == DialogResult.OK &&
!string.IsNullOrEmpty(dialogExport.SelectedPath))
200:                     {
201:                         var outputFileName = dialogExport.SelectedPath + "/" +
202:                         Guid.NewGuid() + ".pdf";
203:                         using (var outputFile = new PdfDocument())
204:                         {

```

```

201:                                     outputFile.Options.CompressContentStreams = true;
202:
203: outputFile.Options.EnableCcittCompressionForBilevelImages = true;
204:                                     outputFile.Options.FlateEncodeMode =
PdfFlateEncodeMode.BestCompression;
205:                                     var fileCount = lbFiles.Items.Count;
206:                                     var combinedFiles = 0;
207:                                     for (int i = 0; i < lbFiles.Items.Count; i++)
208:                                     {
209:                                         try
210:                                         {
211:                                             var inputDocument =
PdfReader.Open(lbFiles.Items[i].ToString(), PdfDocumentOpenMode.Import);
212:                                             var count = inputDocument.PageCount;
213:                                             for (int idx = 0; idx < count; idx++)
214:                                             {
215:                                                 var page = inputDocument.Pages[idx];
216:                                                 outputFile.AddPage(page);
217:                                             }
218:                                             inputDocument.Close();
219:                                             combinedFiles++;
220:                                         }
221:                                         catch (Exception)
222:                                         {
223:                                             fileCount--;
224:                                         }
225:
226:                                     pbFiles.Value = (combinedFiles * 100 / fileCount);
227:                                     if (pbFiles.Value > pbFiles.Maximum)
228:                                     {
229:                                         pbFiles.Value = pbFiles.Maximum;
230:                                         ActiveForm.Text = "%" + pbFiles.Value;
231:                                     }
232:                                     outputFile.Save(outputFileName);
233:                                     outputFile.Close();
234:                                     MessageBox.Show(fileCount +
resource.GetString("CombineFileMessage") + outputFileName, resource.GetString("AppTitle"),
MessageBoxButtons.OK, MessageBoxIcon.Information);
235:                                     }
236:
237:                                     pbFiles.Value = pbFiles.Minimum;
238:                                     ActiveForm.Text = resource.GetString("AppTitle");
239:                                 }
240:                             }
241:                         }
242:                     }
243:                     catch (Exception ex)
244:                     {
245:                         MessageBox.Show(resource.GetString("CombineErrorMessage") +
ex.GetAllMessages(), resource.GetString("AppTitle"), MessageBoxButtons.OK,
MessageBoxIcon.Error);
246:                         pbFiles.Value = pbFiles.Minimum;
247:                         ActiveForm.Text = resource.GetString("AppTitle");
248:                     }
249:                 }
250:
251:                 /// <summary>
252:                 /// This function is used to take list of PDF files in listbox,
253:                 /// Combine them with a single file in location which you choose in folder
dialog
254:                 /// And you can see the progress in progress bar
255:                 /// It uses iTextSharp Nuget Package
256:                 /// </summary>
257:                 /// <param name="sender">The sender info (For example Main Form)</param>
258:                 /// <param name="e">Event Arguments</param>
259:                 private void btnCombineITextSharp_Click(object sender, EventArgs e)
260:                 {
261:                     try

```

```

262:         {
263:             if (lbFiles.Items.Count < 2)
264:                 MessageBox.Show(resource.GetString("CombineWarning"),
resource.GetString("AppTitle"), MessageBoxButtons.OK, MessageBoxIcon.Error);
265:             else
266:             {
267:                 using (var dialogExport = new FolderBrowserDialog())
268:                 {
269:                     var result = dialogExport.ShowDialog();
270:                     if (result == DialogResult.OK &&
!string.IsNullOrEmpty(dialogExport.SelectedPath))
271:                     {
272:                         var outputFileName = dialogExport.SelectedPath + "/" +
Guid.NewGuid() + ".pdf";
273:                         var fileCount = lbFiles.Items.Count;
274:                         var combinedFiles = 0;
275:
276:                         var outputFile = new Document();
277:                         using (FileStream outputFileStream = new
FileStream(outputFileName, FileMode.Create))
278:                         {
279:                             using (var pdfWriter = new
iTextSharp.text.pdf.PdfCopy(outputFile, outputFileStream))
280:                             {
281:                                 if (pdfWriter == null)
282:                                 {
283:                                     return;
284:                                 }
285:                                 pdfWriter.SetFullCompression();
286:                                 outputFile.Open();
287:                                 for (int i = 0; i < lbFiles.Items.Count; i++)
288:                                 {
289:                                     try
290:                                     {
291:                                         var pdfReader = new
iTextSharp.text.pdf.PdfReader(lbFiles.Items[i].ToString());
292:                                         pdfReader.ConsolidateNamedDestinations();
293:                                         for (int j = 1; j <=
pdfReader.NumberOfPages; j++)
294:                                         {
295:                                             var page =
pdfWriter.GetImportedPage(pdfReader, j);
296:                                             pdfWriter.AddPage(page);
297:                                         }
298:                                         pdfReader.Close();
299:                                         combinedFiles++;
300:                                     }
301:                                     catch (Exception)
302:                                     {
303:                                     }
304:                                     fileCount--;
305:                                 }
306:                                 pbFiles.Value = (combinedFiles * 100 /
fileCount);
307:                                 if (pbFiles.Value > pbFiles.Maximum)
308:                                     pbFiles.Value = pbFiles.Maximum;
309:                                 ActiveForm.Text = "%" + pbFiles.Value;
310:                             }
311:                             pdfWriter.Close();
312:                         }
313:                         outputFile.Close();
314:                     }
315:
316:                     MessageBox.Show(fileCount +
resource.GetString("CombineFileMessage") + outputFileName, resource.GetString("AppTitle"),
MessageBoxButtons.OK, MessageBoxIcon.Information);
317:
318:                     pbFiles.Value = pbFiles.Minimum;
319:                     ActiveForm.Text = resource.GetString("AppTitle");

```

```

320:         }
321:     }
322: }
323: }
324:     catch (Exception ex)
325:     {
326:         MessageBox.Show(resource.GetString("CombineErrorMessage") +
ex.GetAllMessages(), resource.GetString("AppTitle"), MessageBoxButtons.OK,
MessageBoxIcon.Error);
327:         pbFiles.Value = pbFiles.Minimum;
328:         ActiveForm.Text = resource.GetString("AppTitle");
329:     }
330: }
331:
332: #endregion
333:
334: #region Delete Item Methods
335:
336:     /// <summary>
337:     /// This function is used to delete files in listbox which are chosen
338:     /// If you press the Delete button in your keyboard
339:     /// The selected files will be deleted from the list
340:     /// </summary>
341:     /// <param name="sender">The sender info (For example Main Form)</param>
342:     /// <param name="e">Event Arguments</param>
343:     private void lbFiles_KeyDown(object sender, KeyEventArgs e)
344:     {
345:         if (e.KeyValue == Keys.Delete.GetHashCode())
346:             DeleteFilesFromListBox();
347:     }
348:
349:     /// <summary>
350:     /// This is used to delete selected items in listbox
351:     /// When menu item Delete is clicked and opened dialog
352:     /// </summary>
353:     /// <param name="sender">The sender info (For example Main Form)</param>
354:     /// <param name="e">Event Arguments</param>
355:     private void menuItemDelete_Click(object sender, EventArgs e)
356:     {
357:         if (lbFiles.Items.Count > 0)
358:         {
359:             if (lbFiles.SelectedItems.Count > 0)
360:             {
361:                 var deleteDialog = new DialogResult();
362:                 deleteDialog =
MessageBox.Show(resource.GetString("DeleteWarning1") + lbFiles.SelectedItems.Count +
resource.GetString("DeleteWarning2"), resource.GetString("AppTitle"),
MessageBoxButtons.YesNo, MessageBoxIcon.Warning);
363:                 if (deleteDialog == DialogResult.Yes)
364:                     DeleteFilesFromListBox();
365:             }
366:             else
367:                 MessageBox.Show(resource.GetString("NoSelectedFile"),
resource.GetString("AppTitle"), MessageBoxButtons.OK, MessageBoxIcon.Error);
368:         }
369:         else
370:             MessageBox.Show(resource.GetString("NoFile"),
resource.GetString("AppTitle"), MessageBoxButtons.OK, MessageBoxIcon.Error);
371:     }
372:
373:     /// <summary>
374:     /// This method is used to Delete selected files from ListBox
375:     /// This is used different ways in this app
376:     /// So we created this as a method
377:     /// </summary>
378:     private void DeleteFilesFromListBox()
379:     {
380:         var allItems = new List<string>();
381:         var removedItems = new List<string>();

```

```

382:         var fileCount = lbFiles.SelectedItems.Count;
383:         for (int i = 0; i < lbFiles.SelectedItems.Count; i++)
384:         {
385:             removedItems.Add(lbFiles.SelectedItems[i].ToString());
386:         }
387:         foreach (var item in lbFiles.Items)
388:         {
389:             if (!removedItems.Any(j => j == item.ToString()))
390:                 allItems.Add(item.ToString());
391:         }
392:         lbFiles.Items.Clear();
393:         foreach (var item in allItems)
394:         {
395:             lbFiles.Items.Add(item);
396:         }
397:         MessageBox.Show(fileCount + resource.GetString("DeleteFileMessage"),
resource.GetString("AppTitle"), MessageBoxButtons.OK, MessageBoxIcon.Information);
398:     }
399:
400:     /// <summary>
401:     /// This function is used to clear the file list in listbox
402:     /// </summary>
403:     /// <param name="sender">The sender info (For example Main Form)</param>
404:     /// <param name="e">Event Arguments</param>
405:     private void btnClearList_Click(object sender, EventArgs e)
406:     {
407:         var fileCount = lbFiles.Items.Count;
408:         lbFiles.Items.Clear();
409:         MessageBox.Show(fileCount + resource.GetString("DeleteFileMessage"),
resource.GetString("AppTitle"), MessageBoxButtons.OK, MessageBoxIcon.Information);
410:     }
411:
412:     #endregion
413:
414:     #region Order Item Methods
415:
416:     /// <summary>
417:     /// This method is used to order elements in listbox ascending by full path
418:     /// </summary>
419:     /// <param name="sender">The sender info (For example Main Form)</param>
420:     /// <param name="e">Event Arguments</param>
421:     private void menuItemOrderByPathAscending_Click(object sender, EventArgs e)
422:     {
423:         if (lbFiles.Items.Count > 0)
424:             lbFiles = SortItemsByPath(lbFiles, true);
425:         else
426:             MessageBox.Show(resource.GetString("NoFile"),
resource.GetString("AppTitle"), MessageBoxButtons.OK, MessageBoxIcon.Error);
427:     }
428:
429:     /// <summary>
430:     /// This method is used to order elements in listbox descending by full path
431:     /// </summary>
432:     /// <param name="sender">The sender info (For example Main Form)</param>
433:     /// <param name="e">Event Arguments</param>
434:     private void menuItemOrderByPathDescending_Click(object sender, EventArgs e)
435:     {
436:         if (lbFiles.Items.Count > 0)
437:             lbFiles = SortItemsByPath(lbFiles, false);
438:         else
439:             MessageBox.Show(resource.GetString("NoFile"),
resource.GetString("AppTitle"), MessageBoxButtons.OK, MessageBoxIcon.Error);
440:     }
441:
442:     /// <summary>
443:     /// This method is used to order elements in listbox ascending by file name
444:     /// </summary>
445:     /// <param name="sender">The sender info (For example Main Form)</param>
446:     /// <param name="e">Event Arguments</param>

```

```

447:         private void menuItemOrderByNameAscending_Click(object sender, EventArgs e)
448:         {
449:             if (lbFiles.Items.Count > 0)
450:                 lbFiles = SortItemsByName(lbFiles, true);
451:             else
452:                 MessageBox.Show(resource.GetString("NoFile"),
resource.GetString("AppTitle"), MessageBoxButtons.OK, MessageBoxIcon.Error);
453:         }
454:
455:         /// <summary>
456:         /// This method is used to order elements in listbox descending by file name
457:         /// </summary>
458:         /// <param name="sender">The sender info (For example Main Form)</param>
459:         /// <param name="e">Event Arguments</param>
460:         private void menuItemOrderByNameDescending_Click(object sender, EventArgs e)
461:         {
462:             if (lbFiles.Items.Count > 0)
463:                 lbFiles = SortItemsByName(lbFiles, false);
464:             else
465:                 MessageBox.Show(resource.GetString("NoFile"),
resource.GetString("AppTitle"), MessageBoxButtons.OK, MessageBoxIcon.Error);
466:         }
467:
468:         /// <summary>
469:         /// This method is used to order items by full path in listbox
470:         /// And return them with the given order as parametre
471:         /// </summary>
472:         /// <param name="listBox">ListBox Info</param>
473:         /// <param name="ascending">Order Type is Ascending Or Not</param>
474:         private static ListBox SortItemsByPath(ListBox listBox, bool ascending)
475:         {
476:             List<object> items;
477:             items = listBox.Items.OfType<object>().ToList();
478:             listBox.Items.Clear();
479:             if (ascending)
480:                 listBox.Items.AddRange(items.OrderBy(i => i).ToArray());
481:             else
482:                 listBox.Items.AddRange(items.OrderByDescending(i => i).ToArray());
483:             return listBox;
484:         }
485:
486:         /// <summary>
487:         /// This method is used to order items by file name in listbox
488:         /// And return them with the given order as parametre
489:         /// </summary>
490:         /// <param name="listBox">ListBox Info</param>
491:         /// <param name="ascending">Order Type is Ascending Or Not</param>
492:         private static ListBox SortItemsByName(ListBox listBox, bool ascending)
493:         {
494:             var fileNameList = new List<string>();
495:             List<object> items;
496:             items = listBox.Items.OfType<object>().ToList();
497:             const char c = '\u005c';
498:             foreach (var item in items)
499:             {
500:                 var fullPath = item.ToString();
501:                 var list = fullPath.Split(c).ToList();
502:                 if (list.Count > 0)
503:                     fileNameList.Add(list[list.Count - 1]);
504:             }
505:
506:             if (ascending)
507:                 fileNameList = fileNameList.OrderBy(j => j).ToList();
508:             else
509:                 fileNameList = fileNameList.OrderByDescending(j => j).ToList();
510:
511:             listBox.Items.Clear();
512:             foreach (var item in fileNameList)
513:                 listBox.Items.Add(items.First(j => j.ToString().EndsWith(item)));

```



```

514:
515:         return listBox;
516:     }
517:
518:     #endregion
519:
520:     #region Item Drag and Drop Methods
521:
522:     /// <summary>
523:     /// This is used to control mouse actions and if that is not right click
524:     /// Start to drag and drop item in listbox
525:     /// </summary>
526:     /// <param name="sender">The sender info (For example Main Form)</param>
527:     /// <param name="e">Event Arguments</param>
528:     private void lbFiles_MouseDown(object sender, MouseEventArgs e)
529:     {
530:         if (!e.Button.Equals(MouseButtons.Right))
531:         {
532:             if (lbFiles.SelectedItem == null) return;
533:             lbFiles.DoDragDrop(lbFiles.SelectedItem, DragDropEffects.Move);
534:         }
535:     }
536:
537:     /// <summary>
538:     /// This is used to give effect while dragging and dropping an item in listbox
539:     /// </summary>
540:     /// <param name="sender">The sender info (For example Main Form)</param>
541:     /// <param name="e">Event Arguments</param>
542:     private void lbFiles_DragOver(object sender, DragEventArgs e)
543:     {
544:         e.Effect = DragDropEffects.Move;
545:     }
546:
547:     /// <summary>
548:     /// This is used to drag and drop an item in listbox
549:     /// Which is used to order books and combine them with that order
550:     /// </summary>
551:     /// <param name="sender">The sender info (For example Main Form)</param>
552:     /// <param name="e">Event Arguments</param>
553:     private void lbFiles_DragDrop(object sender, DragEventArgs e)
554:     {
555:         var point = lbFiles.PointToClient(new Point(e.X, e.Y));
556:         var index = this.lbFiles.IndexFromPoint(point);
557:         if (index < 0) index = this.lbFiles.Items.Count - 1;
558:         // type of string because file names stored in string in listbox
559:         var data = e.Data.GetData(typeof(string));
560:         this.lbFiles.Items.Remove(data);
561:         this.lbFiles.Items.Insert(index, data);
562:     }
563:
564:     #endregion
565: }
566: }

```

### Namespaces

| Name                     | Description                    |
|--------------------------|--------------------------------|
| PdfCombiner (see page 1) | This is namespace PdfCombiner. |

## 1.2.5 FrmMain.Designer.cs

This is file FrmMain.Designer.cs.



## Body Source

```

1: ?namespace PdfCombiner
2: {
3:     partial class FrmMain
4:     {
5:         /// <summary>
6:         /// Required designer variable.
7:         /// </summary>
8:         private System.ComponentModel.IContainer components;
9:
10:        /// <summary>
11:        /// Clean up any resources being used.
12:        /// </summary>
13:        /// <param name="disposing">true if managed resources should be disposed;
otherwise, false.</param>
14:        protected override void Dispose(bool disposing)
15:        {
16:            if (disposing && (components != null))
17:            {
18:                components.Dispose();
19:            }
20:            base.Dispose(disposing);
21:        }
22:
23:        #region Windows Form Designer generated code
24:
25:        /// <summary>
26:        /// Required method for Designer support - do not modify
27:        /// the contents of this method with the code editor.
28:        /// </summary>
29:        private void InitializeComponent()
30:        {
31:            this.components = new System.ComponentModel.Container();
32:            System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(FrmMain));
33:            this.btnCombinePdfSharp = new System.Windows.Forms.Button();
34:            this.btnAddFolder = new System.Windows.Forms.Button();
35:            this.btnAddFile = new System.Windows.Forms.Button();
36:            this.btnClearList = new System.Windows.Forms.Button();
37:            this.lbFiles = new System.Windows.Forms.ListBox();
38:            this.lblDetails = new System.Windows.Forms.Label();
39:            this.pbFiles = new System.Windows.Forms.ProgressBar();
40:            this.btnCombineITextSharp = new System.Windows.Forms.Button();
41:            this.menuStrip = new
System.Windows.Forms.ContextMenuStrip(this.components);
42:            this.menuItemDelete = new System.Windows.Forms.ToolStripMenuItem();
43:            this.menuItemOrderByPathAscending = new
System.Windows.Forms.ToolStripMenuItem();
44:            this.menuItemOrderByPathDescending = new
System.Windows.Forms.ToolStripMenuItem();
45:            this.menuItemOrderByNameAscending = new
System.Windows.Forms.ToolStripMenuItem();
46:            this.menuItemOrderByNameDescending = new
System.Windows.Forms.ToolStripMenuItem();
47:            this.cmbLanguage = new System.Windows.Forms.ComboBox();
48:            this.menuStrip.SuspendLayout();
49:            this.SuspendLayout();
50:            //
51:            // btnCombinePdfSharp
52:            //
53:            this.btnCombinePdfSharp.Image = ((System.Drawing.Image)
(resources.GetObject("btnCombinePdfSharp.Image")));
54:            this.btnCombinePdfSharp.ImageAlign =
System.Drawing.ContentAlignment.MiddleLeft;
55:            this.btnCombinePdfSharp.Location = new System.Drawing.Point(442, 5);
56:            this.btnCombinePdfSharp.Name = "btnCombinePdfSharp";
57:            this.btnCombinePdfSharp.Size = new System.Drawing.Size(214, 37);

```

```

58:         this.btnCombinePdfSharp.TabIndex = 0;
59:         this.btnCombinePdfSharp.Text = "Combine Files (PdfSharp)";
60:         this.btnCombinePdfSharp.TextAlign =
System.Drawing.ContentAlignment.MiddleRight;
61:         this.btnCombinePdfSharp.UseVisualStyleBackColor = true;
62:         this.btnCombinePdfSharp.Click += new
System.EventHandler(this.btnCombinePdfSharp_Click);
63:         //
64:         // btnAddFolder
65:         //
66:         this.btnAddFolder.Image = ((System.Drawing.Image)
(resources.GetObject("btnAddFolder.Image")));
67:         this.btnAddFolder.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;
68:         this.btnAddFolder.Location = new System.Drawing.Point(197, 5);
69:         this.btnAddFolder.Name = "btnAddFolder";
70:         this.btnAddFolder.Size = new System.Drawing.Size(171, 37);
71:         this.btnAddFolder.TabIndex = 2;
72:         this.btnAddFolder.Text = "Add Folder";
73:         this.btnAddFolder.TextAlign = System.Drawing.ContentAlignment.MiddleRight;
74:         this.btnAddFolder.UseVisualStyleBackColor = true;
75:         this.btnAddFolder.Click += new
System.EventHandler(this.btnAddFolder_Click);
76:         //
77:         // btnAddFile
78:         //
79:         this.btnAddFile.Image = ((System.Drawing.Image)
(resources.GetObject("btnAddFile.Image")));
80:         this.btnAddFile.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;
81:         this.btnAddFile.Location = new System.Drawing.Point(12, 5);
82:         this.btnAddFile.Name = "btnAddFile";
83:         this.btnAddFile.Size = new System.Drawing.Size(179, 37);
84:         this.btnAddFile.TabIndex = 3;
85:         this.btnAddFile.Text = "Add Files";
86:         this.btnAddFile.TextAlign = System.Drawing.ContentAlignment.MiddleRight;
87:         this.btnAddFile.UseVisualStyleBackColor = true;
88:         this.btnAddFile.Click += new System.EventHandler(this.btnAddFile_Click);
89:         //
90:         // btnClearList
91:         //
92:         this.btnClearList.Image = ((System.Drawing.Image)
(resources.GetObject("btnClearList.Image")));
93:         this.btnClearList.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;
94:         this.btnClearList.Location = new System.Drawing.Point(12, 45);
95:         this.btnClearList.Name = "btnClearList";
96:         this.btnClearList.Size = new System.Drawing.Size(179, 37);
97:         this.btnClearList.TabIndex = 4;
98:         this.btnClearList.Text = "Clear File List";
99:         this.btnClearList.TextAlign = System.Drawing.ContentAlignment.MiddleRight;
100:        this.btnClearList.UseVisualStyleBackColor = true;
101:        this.btnClearList.Click += new
System.EventHandler(this.btnClearList_Click);
102:        //
103:        // lbFiles
104:        //
105:        this.lbFiles.AllowDrop = true;
106:        this.lbFiles.FormattingEnabled = true;
107:        this.lbFiles.Location = new System.Drawing.Point(12, 88);
108:        this.lbFiles.Name = "lbFiles";
109:        this.lbFiles.SelectionMode =
System.Windows.Forms.SelectionMode.MultiExtended;
110:        this.lbFiles.Size = new System.Drawing.Size(644, 186);
111:        this.lbFiles.TabIndex = 5;
112:        this.lbFiles.DragDrop += new
System.Windows.Forms.DragEventHandler(this.lbFiles_DragDrop);
113:        this.lbFiles.DragOver += new
System.Windows.Forms.DragEventHandler(this.lbFiles_DragOver);
114:        this.lbFiles.KeyDown += new
System.Windows.Forms.KeyEventHandler(this.lbFiles_KeyDown);
115:        this.lbFiles.MouseDown += new

```

```

System.Windows.Forms.MouseEventHandler(this.lbFiles_MouseDown);
116:         //
117:         // lblDetails
118:         //
119:         this.lblDetails.AutoSize = true;
120:         this.lblDetails.Location = new System.Drawing.Point(12, 65);
121:         this.lblDetails.Name = "lblDetails";
122:         this.lblDetails.Size = new System.Drawing.Size(0, 13);
123:         this.lblDetails.TabIndex = 6;
124:         //
125:         // pbFiles
126:         //
127:         this.pbFiles.Location = new System.Drawing.Point(197, 45);
128:         this.pbFiles.Name = "pbFiles";
129:         this.pbFiles.Size = new System.Drawing.Size(239, 37);
130:         this.pbFiles.TabIndex = 7;
131:         //
132:         // btnCombineITextSharp
133:         //
134:         this.btnCombineITextSharp.Image = ((System.Drawing.Image)
(resources.GetObject("btnCombineITextSharp.Image")));
135:         this.btnCombineITextSharp.ImageAlign =
System.Drawing.ContentAlignment.MiddleLeft;
136:         this.btnCombineITextSharp.Location = new System.Drawing.Point(442, 45);
137:         this.btnCombineITextSharp.Name = "btnCombineITextSharp";
138:         this.btnCombineITextSharp.Size = new System.Drawing.Size(214, 37);
139:         this.btnCombineITextSharp.TabIndex = 9;
140:         this.btnCombineITextSharp.Text = "Combine Files (iTextSharp)";
141:         this.btnCombineITextSharp.TextAlign =
System.Drawing.ContentAlignment.MiddleRight;
142:         this.btnCombineITextSharp.UseVisualStyleBackColor = true;
143:         this.btnCombineITextSharp.Click += new
System.EventHandler(this.btnCombineITextSharp_Click);
144:         //
145:         // menuStrip
146:         //
147:         this.menuStrip.Items.AddRange(new System.Windows.Forms.ToolStripItem[]
{this.menuItemDelete, this.menuItemOrderByPathAscending,
this.menuItemOrderByPathDescending, this.menuItemOrderByNameAscending,
this.menuItemOrderByNameDescending});
148:         this.menuStrip.Name = "menuStrip";
149:         this.menuStrip.Size = new System.Drawing.Size(229, 114);
150:         //
151:         // menuItemDelete
152:         //
153:         this.menuItemDelete.Image = ((System.Drawing.Image)
(resources.GetObject("menuItemDelete.Image")));
154:         this.menuItemDelete.Name = "menuItemDelete";
155:         this.menuItemDelete.Size = new System.Drawing.Size(228, 22);
156:         this.menuItemDelete.Text = "Delete";
157:         this.menuItemDelete.Click += new
System.EventHandler(this.menuItemDelete_Click);
158:         //
159:         // menuItemOrderByPathAscending
160:         //
161:         this.menuItemOrderByPathAscending.Image = ((System.Drawing.Image)
(resources.GetObject("menuItemOrderByPathAscending.Image")));
162:         this.menuItemOrderByPathAscending.Name = "menuItemOrderByPathAscending";
163:         this.menuItemOrderByPathAscending.Size = new System.Drawing.Size(228, 22);
164:         this.menuItemOrderByPathAscending.Text = "Order By Path (Ascending)";
165:         this.menuItemOrderByPathAscending.Click += new
System.EventHandler(this.menuItemOrderByPathAscending_Click);
166:         //
167:         // menuItemOrderByPathDescending
168:         //
169:         this.menuItemOrderByPathDescending.Image = ((System.Drawing.Image)
(resources.GetObject("menuItemOrderByPathDescending.Image")));
170:         this.menuItemOrderByPathDescending.Name = "menuItemOrderByPathDescending";
171:         this.menuItemOrderByPathDescending.Size = new System.Drawing.Size(228, 22);

```

```

172:         this.menuItemOrderByPathDescending.Text = "Order By Path (Descending)";
173:         this.menuItemOrderByPathDescending.Click += new
System.EventHandler(this.menuItemOrderByPathDescending_Click);
174:         //
175:         // menuItemOrderByNameAscending
176:         //
177:         this.menuItemOrderByNameAscending.Image = ((System.Drawing.Image)
(resources.GetObject("menuItemOrderByNameAscending.Image")));
178:         this.menuItemOrderByNameAscending.Name = "menuItemOrderByNameAscending";
179:         this.menuItemOrderByNameAscending.Size = new System.Drawing.Size(228, 22);
180:         this.menuItemOrderByNameAscending.Text = "Order By Name (Ascending)";
181:         this.menuItemOrderByNameAscending.Click += new
System.EventHandler(this.menuItemOrderByNameAscending_Click);
182:         //
183:         // menuItemOrderByNameDescending
184:         //
185:         this.menuItemOrderByNameDescending.Image = ((System.Drawing.Image)
(resources.GetObject("menuItemOrderByNameDescending.Image")));
186:         this.menuItemOrderByNameDescending.Name = "menuItemOrderByNameDescending";
187:         this.menuItemOrderByNameDescending.Size = new System.Drawing.Size(228, 22);
188:         this.menuItemOrderByNameDescending.Text = "Order By Name (Descending)";
189:         this.menuItemOrderByNameDescending.Click += new
System.EventHandler(this.menuItemOrderByNameDescending_Click);
190:         //
191:         // cmbLanguage
192:         //
193:         this.cmbLanguage.Anchor = System.Windows.Forms.AnchorStyles.None;
194:         this.cmbLanguage.FormattingEnabled = true;
195:         this.cmbLanguage.Items.AddRange(new object[] { "TR", "EN", "DE", "FR",
"RU", "ES" });
196:         this.cmbLanguage.Location = new System.Drawing.Point(374, 12);
197:         this.cmbLanguage.Name = "cmbLanguage";
198:         this.cmbLanguage.Size = new System.Drawing.Size(62, 21);
199:         this.cmbLanguage.TabIndex = 10;
200:         this.cmbLanguage.Text = "Select Language";
201:         this.cmbLanguage.SelectedIndexChanged += new
System.EventHandler(this.cmbLanguage_SelectedIndexChanged);
202:         //
203:         // FrmMain
204:         //
205:         this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
206:         this.AutoScaleMode = System.Windows.Forms.AutoScaleModeMode.Font;
207:         this.ClientSize = new System.Drawing.Size(668, 287);
208:         this.Controls.Add(this.cmbLanguage);
209:         this.Controls.Add(this.btnCombineITextSharp);
210:         this.Controls.Add(this.pbFiles);
211:         this.Controls.Add(this.lblDetails);
212:         this.Controls.Add(this.lbFiles);
213:         this.Controls.Add(this.btnClearList);
214:         this.Controls.Add(this.btnAddFile);
215:         this.Controls.Add(this.btnAddFolder);
216:         this.Controls.Add(this.btnCombinePdfSharp);
217:         this.Name = "FrmMain";
218:         this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
219:         this.Text = "PDF Combiner";
220:         this.FormClosed += new
System.Windows.Forms.FormClosedEventHandler(this.FrmMain_FormClosed);
221:         this.Shown += new System.EventHandler(this.FrmMain_Shown);
222:         this.menuStrip.ResumeLayout(false);
223:         this.ResumeLayout(false);
224:         this.PerformLayout();
225:     }
226:
227:     public System.Windows.Forms.ComboBox cmbLanguage;
228:
229:     #endregion
230:
231:     private System.Windows.Forms.Button btnCombinePdfSharp;
232:     private System.Windows.Forms.Button btnAddFolder;

```

```
233:         private System.Windows.Forms.Button btnAddFile;
234:         private System.Windows.Forms.Button btnClearList;
235:         private System.Windows.Forms.ListBox lbFiles;
236:         private System.Windows.Forms.Label lblDetails;
237:         private System.Windows.Forms.ProgressBar pbFiles;
238:         private System.Windows.Forms.Button btnCombineITextSharp;
239:         private System.Windows.Forms.ContextMenuStrip menuStrip;
240:         private System.Windows.Forms.ToolStripMenuItem menuItemDelete;
241:         private System.Windows.Forms.ToolStripMenuItem menuItemOrderByPathAscending;
242:         private System.Windows.Forms.ToolStripMenuItem menuItemOrderByPathDescending;
243:         private System.Windows.Forms.ToolStripMenuItem menuItemOrderByNameAscending;
244:         private System.Windows.Forms.ToolStripMenuItem menuItemOrderByNameDescending;
245:     }
246: }
247:
```

Namespaces

| Name                                       | Description                    |
|--|--------------------------------|
| PdfCombiner ( <a href="#">see page 1</a> ) | This is namespace PdfCombiner. |

## 1.2.6 PdfCombiner.csproj

This is file PdfCombiner.csproj.

## 1.2.7 PdfCombiner.sln

This is file PdfCombiner.sln.

## 1.2.8 Program.cs

This is file Program.cs.

Body Source

```
1: ?using System;
2: using System.Collections.Generic;
3: using System.Linq;
4: using System.Windows.Forms;
5:
6: namespace PdfCombiner
7: {
8:     static class Program
9:     {
10:         /// <summary>
11:         /// The main entry point for the application.
12:         /// </summary>
13:         [STAThread]
14:         static void Main()
15:         {
16:             Application.EnableVisualStyles();
17:             Application.SetCompatibleTextRenderingDefault(false);
18:             Application.Run(new FrmMain());
19:         }
20:     }
21: }
```

Namespaces

| Name                      | Description                    |
|---------------------------|--------------------------------|
| PdfCombiner ( see page 1) | This is namespace PdfCombiner. |

# 1.2.9 Resources.Designer.cs

This code was generated by a tool. Runtime Version:4.0.30319.42000

Changes to this file may cause incorrect behavior and will be lost if the code is regenerated.

Body Source

```
1: ?//-----
2: // <auto-generated>
3: //   This code was generated by a tool.
4: //   Runtime Version:4.0.30319.42000
5: //
6: //   Changes to this file may cause incorrect behavior and will be lost if
7: //   the code is regenerated.
8: // </auto-generated>
9: //-----
10:
11: namespace PdfCombiner.Properties {
12:     using System;
13:
14:
15:     /// <summary>
16:     ///   A strongly-typed resource class, for looking up localized strings, etc.
17:     /// </summary>
18:     // This class was auto-generated by the StronglyTypedResourceBuilder
19:     // class via a tool like ResGen or Visual Studio.
20:     // To add or remove a member, edit your .ResX file then rerun ResGen
21:     // with the /str option, or rebuild your VS project.
22:
23: [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Resources.Tools.StronglyTypedResourceBuilder",
24: "15.0.0.0")]
25: [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
26: [global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]
27:     internal class Resources {
28:
29:         private static global::System.Resources.ResourceManager resourceMan;
30:
31:         private static global::System.Globalization.CultureInfo resourceCulture;
32:
33: [global::System.Diagnostics.CodeAnalysis.SuppressMessageAttribute("Microsoft.Performance",
34: "CA1811:AvoidUncalledPrivateCode")]
35:         internal Resources() {
36:
37:         }
38:
39:         /// <summary>
40:         ///   Returns the cached ResourceManager instance used by this class.
41:         /// </summary>
42:
43: [global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableState.Advanced)]
44:         internal static global::System.Resources.ResourceManager ResourceManager {
45:             get {
46:                 if (object.ReferenceEquals(resourceMan, null)) {
47:                     global::System.Resources.ResourceManager temp = new
48: global::System.Resources.ResourceManager("PdfCombiner.Properties.Resources",
49: typeof(Resources).Assembly);
```

```
43:         resourceMan = temp;
44:     }
45:     return resourceMan;
46: }
47:
48:
49:     /// <summary>
50:     ///     Overrides the current thread's CurrentUICulture property for all
51:     ///     resource lookups using this strongly typed resource class.
52:     /// </summary>
53:
54: [global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableState.Advanced)]
55:     internal static global::System.Globalization.CultureInfo Culture {
56:     get {
57:         return resourceCulture;
58:     }
59:     set {
60:         resourceCulture = value;
61:     }
62:
63:     /// <summary>
64:     ///     Looks up a localized string similar to PDF Birlestirici.
65:     /// </summary>
66:     internal static string AppTitle {
67:     get {
68:         return ResourceManager.GetString("AppTitle", resourceCulture);
69:     }
70: }
71: }
72: }
```

Namespaces

| Name                     | Description                    |
|--------------------------|--------------------------------|
| PdfCombiner (see page 1) | This is namespace PdfCombiner. |

# 1.2.10 Settings.Designer.cs

This code was generated by a tool. Runtime Version:4.0.30319.42000

Changes to this file may cause incorrect behavior and will be lost if the code is regenerated.

Body Source

```
1: ?//-----
2: // <auto-generated>
3: //     This code was generated by a tool.
4: //     Runtime Version:4.0.30319.42000
5: //
6: //     Changes to this file may cause incorrect behavior and will be lost if
7: //     the code is regenerated.
8: // </auto-generated>
9: //-----
10:
11: namespace PdfCombiner.Properties
12: {
13:
14:     [global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]
15:
16: [global::System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.VisualStudio.Editors.SettingsDesigner.SettingsSingleFileGenerator",
17: "11.0.0.0")]
18:     internal sealed partial class Settings :
```



```
global::System.Configuration.ApplicationSettingsBase
17:     {
18:
19:         private static Settings defaultInstance =
((Settings)(global::System.Configuration.ApplicationSettingsBase.Synchronized(new
Settings())));
20:     }
21: }
```

Namespaces

| Name                       | Description                    |
|----------------------------|--------------------------------|
| PdfCombiner (🔗 see page 1) | This is namespace PdfCombiner. |



# Index

## A

AssemblyInfo.cs 27

## B

BaseException class 4

about BaseException class 4

BaseException 5

BaseException.cs 28

## E

ExceptionExtensions class 5

about ExceptionExtensions class 5

GetAllMessages 6

ExceptionExtensions.cs 29

## F

Files 27

FrmMain class 6

about FrmMain class 6

btnAddFile 8

btnAddFile\_Click 11

btnAddFolder 9

btnAddFolder\_Click 12

btnClearList 9

btnClearList\_Click 12

btnCombineITextSharp 9

btnCombineITextSharp\_Click 13

btnCombinePdfSharp 9

btnCombinePdfSharp\_Click 14

cmbLanguage 9

cmbLanguage\_SelectedIndexChanged 15

components 9

DeleteFilesFromListBox 16

Dispose 16

FormMembersNameInitialization 17

FrmMain 8

FrmMain\_FormClosed 17

FrmMain\_Shown 18

InitializeComponent 18

InitializeFileDialog 21

lbFiles 9

lbFiles\_DragDrop 22

lbFiles\_DragOver 22

lbFiles\_KeyDown 22

lbFiles\_MouseDown 23

lblDetails 10

menuItemDelete 10

menuItemDelete\_Click 23

menuItemOrderByNameAscending 10

menuItemOrderByNameAscending\_Click 24

menuItemOrderByNameDescending 10

menuItemOrderByNameDescending\_Click 24

menuItemOrderByPathAscending 10

menuItemOrderByPathAscending\_Click 24

menuItemOrderByPathDescending 10

menuItemOrderByPathDescending\_Click 25

menuStrip 10

pbFiles 11

resource 11

SortItemsByName 25

SortItemsByPath 26

FrmMain.cs 29

FrmMain.Designer.cs 38

## P

PdfCombiner 1

PdfCombiner namespace 1

Classes 4

PdfCombiner.BaseException 4

PdfCombiner.BaseException.BaseException 5

PdfCombiner.csproj 43

PdfCombiner.ExceptionExtensions 5

PdfCombiner.ExceptionExtensions.GetAllMessages 6

PdfCombiner.FrmMain 6

PdfCombiner.FrmMain.btnAddFile 8

PdfCombiner.FrmMain.btnAddFile\_Click 11

PdfCombiner.FrmMain.btnAddFolder 9

|  |  |
|--|--|
| PdfCombiner.FrmMain.btnAddFolder_Click 12                  | PdfCombiner.FrmMain.menuStrip 10                   |
| PdfCombiner.FrmMain.btnClearList 9                         | PdfCombiner.FrmMain.pbFiles 11                     |
| PdfCombiner.FrmMain.btnClearList_Click 12                  | PdfCombiner.FrmMain.resource 11                    |
| PdfCombiner.FrmMain.btnCombineITextSharp 9                 | PdfCombiner.FrmMain.SortItemsByName 25             |
| PdfCombiner.FrmMain.btnCombineITextSharp_Click 13          | PdfCombiner.FrmMain.SortItemsByPath 26             |
| PdfCombiner.FrmMain.btnCombinePdfSharp 9                   | PdfCombiner.Program 26                             |
| PdfCombiner.FrmMain.btnCombinePdfSharp_Click 14            | PdfCombiner.Program.Main 27                        |
| PdfCombiner.FrmMain.cmbLanguage 9                          | PdfCombiner.Properties 1                           |
| PdfCombiner.FrmMain.cmbLanguage_SelectedIndexChanged 15    | PdfCombiner.Properties namespace 1                 |
| PdfCombiner.FrmMain.components 9                           | Classes 1  |
| PdfCombiner.FrmMain.DeleteFilesFromListBox 16              | PdfCombiner.Properties.Resources 1                 |
| PdfCombiner.FrmMain.Dispose 16                             | PdfCombiner.Properties.Resources.AppTitle 3        |
| PdfCombiner.FrmMain.FormMembersNameInitialization 17       | PdfCombiner.Properties.Resources.Culture 3         |
| PdfCombiner.FrmMain.FrmMain 8                              | PdfCombiner.Properties.Resources.resourceCulture 2 |
| PdfCombiner.FrmMain.FrmMain_FormClosed 17                  | PdfCombiner.Properties.Resources.resourceMan 3     |
| PdfCombiner.FrmMain.FrmMain_Shown 18                       | PdfCombiner.Properties.Resources.ResourceManager 3 |
| PdfCombiner.FrmMain.InitializeComponent 18                 | PdfCombiner.Properties.Resources.Resources 2       |
| PdfCombiner.FrmMain.InitializeFileDialog 21                | PdfCombiner.Properties.Settings 3                  |
| PdfCombiner.FrmMain.lbFiles 9                              | PdfCombiner.Properties.Settings.defaultInstance 4  |
| PdfCombiner.FrmMain.lbFiles_DragDrop 22                    | PdfCombiner.sln 43                                 |
| PdfCombiner.FrmMain.lbFiles_DragOver 22                    | Program class 26                                   |
| PdfCombiner.FrmMain.lbFiles_KeyDown 22                     | about Program class 26                             |
| PdfCombiner.FrmMain.lbFiles_MouseDown 23                   | Main 27  |
| PdfCombiner.FrmMain.lblDetails 10                          | Program.cs 43                                      |
| PdfCombiner.FrmMain.menuItemDelete 10                      |  |
| PdfCombiner.FrmMain.menuItemDelete_Click 23                | <b>R</b>   |
| PdfCombiner.FrmMain.menuItemOrderByNameAscending 10        | Resources class 1                                  |
| PdfCombiner.FrmMain.menuItemOrderByNameAscending_Click 24  | about Resources class 1                            |
| PdfCombiner.FrmMain.menuItemOrderByNameDescending 10       | AppTitle 3   |
| PdfCombiner.FrmMain.menuItemOrderByNameDescending_Click 24 | Culture 3  |
| PdfCombiner.FrmMain.menuItemOrderByPathAscending 10        | resourceCulture 2                                  |
| PdfCombiner.FrmMain.menuItemOrderByPathAscending_Click 24  | resourceMan 3                                      |
| PdfCombiner.FrmMain.menuItemOrderByPathDescending 10       | ResourceManager 3                                  |
| PdfCombiner.FrmMain.menuItemOrderByPathDescending_Click 25 | Resources 2  |
|  | Resources.Designer.cs 44                           |
|  | <b>S</b>   |
|  | Settings class 3                                   |
|  | about Settings class 3                             |
|  | defaultInstance 4                                  |

Settings.Designer.cs 45