

PDF Combiner

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

Table of Contents

Symbol Reference	1
PdfCombiner Namespace	1
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	27
Program Methods	27
Files	28
AssemblyInfo.cs	28
BaseException.cs	29
ExceptionExtensions.cs	29
FrmMain.cs	30
FrmMain.Designer.cs	40
PdfCombiner.csproj	44
PdfCombiner.sln	44
Program.cs	45
Resources.Designer.cs	45
Settings.Designer.cs	47
Index	a

1 Symbol Reference





1.1 PdfCombiner Namespace

This is namespace PdfCombiner.

Namespaces

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



Classes

	Name	Description
	BaseException (see page 4)	This function is used to take exceptions which can be thrown in PDF Combiner App
	ExceptionExtensions (see page 5)	This is class PdfCombiner.ExceptionExtensions.
	FrmMain (see page 6)	This is class PdfCombiner.FrmMain.
	Program (see page 27)	This is class PdfCombiner.Program.

1.1.1 PdfCombiner.Properties Namespace

This is namespace PdfCombiner.Properties.



Classes

	Name	Description
	Resources (see page 1)	A strongly-typed resource class, for looking up localized strings, etc.
	Settings (see page 3)	This is class PdfCombiner.Properties.Settings.

1.1.1.1 Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	Resources (see page 1)	A strongly-typed resource class, for looking up localized strings, etc.
	Settings (see page 3)	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 45](#))



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#

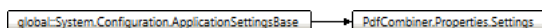
```
[global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableState.Advanced)]
internal static global::System.Globalization.CultureInfo Culture;
```

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 47)

Description

This is class PdfCombiner.Properties.Settings.

Settings Fields

	Name	Description
	defaultInstance (see page 4)	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

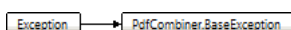
The following table lists classes in this documentation.

Classes

	Name	Description
	BaseException (see page 4)	This function is used to take exceptions which can be thrown in PDF Combiner App
	ExceptionExtensions (see page 5)	This is class PdfCombiner.ExceptionExtensions.
	FrmMain (see page 6)	This is class PdfCombiner.FrmMain.
	Program (see page 27)	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 29](#))

Methods

	Name	Description
	BaseException (see page 5)	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

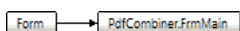
Parameters	Description
this Exception exception	Exception

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 40)





















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 (see page 8)	This is btnAddFile, a member of class FrmMain.
	btnAddFolder (see page 9)	This is btnAddFolder, a member of class FrmMain.
	btnClearList (see page 9)	This is btnClearList, a member of class FrmMain.
	btnCombineITextSharp (see page 9)	This is btnCombineITextSharp, a member of class FrmMain.
	btnCombinePdfSharp (see page 9)	This is btnCombinePdfSharp, a member of class FrmMain.
	cmbLanguage (see page 9)	This is cmbLanguage, a member of class FrmMain.
	components (see page 9)	Required designer variable.
 	languageList (see page 9)	Culture lists whose resources have been added to project
	lbFiles (see page 10)	This is lbFiles, a member of class FrmMain.
	lblDetails (see page 10)	This is lblDetails, a member of class FrmMain.
	menuItemDelete (see page 10)	This is menuItemDelete, a member of class FrmMain.
	menuItemOrderByNameAscending (see page 10)	This is menuItemOrderByNameAscending, a member of class FrmMain.
	menuItemOrderByNameDescending (see page 10)	This is menuItemOrderByNameDescending, a member of class FrmMain.
	menuItemOrderByPathAscending (see page 10)	This is menuItemOrderByPathAscending, a member of class FrmMain.
	menuItemOrderByPathDescending (see page 11)	This is menuItemOrderByPathDescending, a member of class FrmMain.
	menuStrip (see page 11)	This is menuStrip, a member of class FrmMain.
	pbFiles (see page 11)	This is pbFiles, a member of class FrmMain.
 	resource (see page 11)	This is resource, a member of class FrmMain.

FrmMain Methods

	Name	Description
	btnAddFile_Click (see page 11)	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 (see page 12)	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 (see page 12)	This function is used to clear the file list in listbox
	btnCombineITextSharp_Click (see page 13)	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 (see page 14)	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 (see page 16)	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
	DeleteFilesFromListBox (see page 16)	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 (see page 17)	Clean up any resources being used.

	FormMembersNameInitialization (see page 17)	This method is used to change text of form members by the Resource file which is selected
	FrmMain_FormClosed (see page 18)	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 22)	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 23)	This is used to give effect while dragging and dropping an item in listbox
	lbFiles_KeyDown (see page 23)	This function is used to operate some process in files in listbox which are chosen
	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 24)	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 25)	This method is used to order elements in listbox descending by file name
	menuItemOrderByPathAscending_Click (see page 25)	This method is used to order elements in listbox ascending by full path
	menuItemOrderByPathDescending_Click (see page 26)	This method is used to order elements in listbox descending by full path
 	SortItemsByName (see page 26)	This method is used to order items by file name in listbox And return them with the given order as parametre
 	SortItemsByPath (see page 27)	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.languageList Field**C#**

```
private static readonly List<string> languageList = new List<string> { "EN", "TR", "DE",
```

```
"FR", "RU", "ES", "IT", "ZH", "AR"};
```

Description

Culture lists whose resources have been added to project

1.1.2.3.2.9 FrmMain.lbFiles Field**C#**

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

Description

This is lbFiles, a member of class FrmMain.

1.1.2.3.2.10 FrmMain.lblDetails Field**C#**

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

Description

This is lblDetails, a member of class FrmMain.

1.1.2.3.2.11 FrmMain.menuItemDelete Field**C#**

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

Description

This is menuItemDelete, a member of class FrmMain.

1.1.2.3.2.12 FrmMain.menuItemOrderByNameAscending Field**C#**

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

Description

This is menuItemOrderByNameAscending, a member of class FrmMain.

1.1.2.3.2.13 FrmMain.menuItemOrderByNameDescending Field**C#**

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

Description

This is menuItemOrderByNameDescending, a member of class FrmMain.

1.1.2.3.2.14 FrmMain.menuItemOrderByPathAscending Field**C#**

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

Description

This is menuItemOrderByPathAscending, a member of class FrmMain.

1.1.2.3.2.15 FrmMain.menuItemOrderByPathDescending Field

C#

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

Description

This is menuItemOrderByPathDescending, a member of class FrmMain.

1.1.2.3.2.16 FrmMain.menuStrip Field

C#

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

Description

This is menuStrip, a member of class FrmMain.

1.1.2.3.2.17 FrmMain.pbFiles Field

C#

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

Description

This is pbFiles, a member of class FrmMain.

1.1.2.3.2.18 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:
19:             MessageBox.Show(addedFileCount + resource.GetString("FileAddMessage"),
20:                 resource.GetString("AppTitle"), MessageBoxButtons.OK,
21:                 MessageBoxIcon.Information);
22:         }
23: }

```

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 &&
7:             !string.IsNullOrEmpty(dialogAddFolder.SelectedPath))
8:         {
9:             var fileNames = Directory.GetFiles(dialogAddFolder.SelectedPath, "*.pdf",
10:                 SearchOption.AllDirectories);
11:             var addedFileCount = 0;
12:             foreach (var file in fileNames)
13:             {
14:                 if (!lbFiles.Items.Contains(file))
15:                 {
16:                     lbFiles.Items.Add(file);
17:                     addedFileCount++;
18:                 }
19:             }
20:
21:             MessageBox.Show(addedFileCount + resource.GetString("FileAddMessage"),
22:                 resource.GetString("AppTitle"), MessageBoxButtons.OK,
23:                 MessageBoxIcon.Information);
24:         }
25:     }
26: }

```

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"),
6: resource.GetString("AppTitle"),
7:     MessageBoxButtons.OK, MessageBoxIcon.Information);
8: }
```

1.1.2.3.4 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"),
7: resource.GetString("AppTitle"),
8:             MessageBoxButtons.OK, MessageBoxIcon.Error);
9:         else
10:        {
11:            using (var dialogExport = new FolderBrowserDialog())
12:            {
13:                var result = dialogExport.ShowDialog();
14:                if (result == DialogResult.OK &&
15: !string.IsNullOrEmpty(dialogExport.SelectedPath))
16:                {
17:                    var outputFileName = dialogExport.SelectedPath + "/" +
18: Guid.NewGuid() + ".pdf";
19:                    var fileCount = lbFiles.Items.Count;
20:                    var combinedFiles = 0;
21:
22:                    var outputFile = new Document();
23:                    using (FileStream outputFileStream = new FileStream(outputFileName,
24: FileMode.Create))
25:                    {
26:                        using (var pdfWriter = new
27: iTextSharp.text.pdf.PdfCopy(outputFile, outputFileStream))
28:                        {
29:                            if (pdfWriter == null)
30:                            {
31:                                // ...
32:                            }
33:                        }
34:                    }
35:                }
36:            }
37:        }
38:    }
39: }
```

```

26:         return;
27:     }
28:
29:     pdfWriter.SetFullCompression();
30:     outputFile.Open();
31:     for (int i = 0; i < lbFiles.Items.Count; i++)
32:     {
33:         try
34:         {
35:             var pdfReader =
36:                 new
iTextSharp.text.pdf.PdfReader(lbFiles.Items[i].ToString());
37:             pdfReader.ConsolidateNamedDestinations();
38:             for (int j = 1; j <= pdfReader.NumberOfPages; j++)
39:             {
40:                 var page = pdfWriter.GetImportedPage(pdfReader,
j);
41:                 pdfWriter.AddPage(page);
42:             }
43:
44:             pdfReader.Close();
45:             combinedFiles++;
46:         }
47:         catch (Exception)
48:         {
49:             fileCount--;
50:         }
51:
52:         pbFiles.Value = (combinedFiles * 100 / fileCount);
53:         if (pbFiles.Value > pbFiles.Maximum)
54:             pbFiles.Value = pbFiles.Maximum;
55:         ActiveForm.Text = "%" + pbFiles.Value;
56:     }
57:
58:     pdfWriter.Close();
59: }
60:
61:     outputFile.Close();
62: }
63:
64:     MessageBox.Show(fileCount +
resource.GetString("CombineFileMessage") + outputFile.Name,
65:         resource.GetString("AppTitle"), MessageBoxButtons.OK,
MessageBoxIcon.Information);
66:
67:     pbFiles.Value = pbFiles.Minimum;
68:     ActiveForm.Text = resource.GetString("AppTitle");
69: }
70: }
71: }
72: }
73: catch (Exception ex)
74: {
75:     MessageBox.Show(resource.GetString("CombineErrorMessage") + ex.GetAllMessages(),
76:         resource.GetString("AppTitle"), MessageBoxButtons.OK, MessageBoxIcon.Error);
77:     pbFiles.Value = pbFiles.Minimum;
78:     ActiveForm.Text = resource.GetString("AppTitle");
79: }
80: }

```

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"),
7:                 MessageBoxButtons.OK, MessageBoxIcon.Error);
8:         else
9:         {
10:            using (var dialogExport = new FolderBrowserDialog())
11:            {
12:                var result = dialogExport.ShowDialog();
13:                if (result == DialogResult.OK &&
!string.IsNullOrEmpty(dialogExport.SelectedPath))
14:                {
15:                    var outputFileName = dialogExport.SelectedPath + "/" +
Guid.NewGuid() + ".pdf";
16:                    using (var outputFile = new PdfDocument())
17:                    {
18:                        outputFile.Options.CompressContentStreams = true;
19:                        outputFile.Options.EnableCcittCompressionForBilevelImages =
true;
20:                        outputFile.Options.FlateEncodeMode =
PdfFlateEncodeMode.BestCompression;
21:                        var fileCount = lbFiles.Items.Count;
22:                        var combinedFiles = 0;
23:
24:                        for (int i = 0; i < lbFiles.Items.Count; i++)
25:                        {
26:                            try
27:                            {
28:                                var inputDocument =
PdfReader.Open(lbFiles.Items[i].ToString(),
29:                                    PdfDocumentOpenMode.Import);
30:                                var count = inputDocument.PageCount;
31:                                for (int idx = 0; idx < count; idx++)
32:                                {
33:                                    var page = inputDocument.Pages[idx];
34:                                    outputFile.AddPage(page);
35:                                }
36:
37:                                inputDocument.Close();
38:                                combinedFiles++;
39:                            }
40:                            catch (Exception)
41:                            {
42:                                fileCount--;
43:                            }
44:
45:                            pbFiles.Value = (combinedFiles * 100 / fileCount);
46:                            if (pbFiles.Value > pbFiles.Maximum)
47:                                pbFiles.Value = pbFiles.Maximum;
48:                            ActiveForm.Text = "%" + pbFiles.Value;
49:                        }
50:
51:                        outputFile.Save(outputFileName);
52:                        outputFile.Close();
53:
54:                        MessageBox.Show(fileCount +

```

```

resource.GetString("CombineFileMessage") + outputFileName,
55:             resource.GetString("AppTitle"), MessageBoxButtons.OK,
MessageBoxIcon.Information);
56:         }
57:     }
58:         pbFiles.Value = pbFiles.Minimum;
59:         ActiveForm.Text = resource.GetString("AppTitle");
60:     }
61: }
62: }
63: }
64: catch (Exception ex)
65: {
66:     MessageBox.Show(resource.GetString("CombineErrorMessage") + ex.GetAllMessages(),
67:         resource.GetString("AppTitle"), MessageBoxButtons.OK, MessageBoxIcon.Error);
68:     pbFiles.Value = pbFiles.Minimum;
69:     ActiveForm.Text = resource.GetString("AppTitle");
70: }
71: }

```

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(
8:                 "PdfCombiner.Resources.AppResources-" +
9:                 cmbLanguage.SelectedItem.ToString().ToLower(new CultureInfo("en-US")),
10:                 Assembly.GetExecutingAssembly());
11:             FormMembersNameInitialization();
12:         }
13:         catch (Exception)
14:         {
15:             resource = new ResourceManager("PdfCombiner.Resources.AppResources-en",
16:                 Assembly.GetExecutingAssembly());
17:             FormMembersNameInitialization();
18:         }
19:     }
20: }

```

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:
11:     foreach (var item in lbFiles.Items)
12:     {
13:         if (!removedItems.Any(j => j == item.ToString()))
14:             allItems.Add(item.ToString());
15:     }
16:
17:     lbFiles.Items.Clear();
18:     foreach (var item in allItems)
19:     {
20:         lbFiles.Items.Add(item);
21:     }
22:
23:     MessageBox.Show(fileCount + resource.GetString("DeleteFileMessage"),
resource.GetString("AppTitle"),
24:         MessageBoxButtons.OK, MessageBoxIcon.Information);
25: }

```

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,
5:         MessageBoxIcon.Information);
6:     Application.Exit();
7: }

```

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:     cmbLanguage.Items.Clear();
6:     foreach (var item in languageList)
7:         cmbLanguage.Items.Add(item);
8:     resource = new ResourceManager("PdfCombiner.Resources.AppResources-tr",
9:         Assembly.GetExecutingAssembly());
10:    FormMembersNameInitialization();
11: }

```

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 operate some process in files in listbox which are chosen

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:     // this is used to delete selected files in listbox when you press Delete key
4:     if (e.KeyValue == Keys.Delete.GetHashCode())
5:         DeleteFilesFromListBox();
6:     // This is used to select all items in listbox when you press Ctrl + A key
combination
7:     if (e.Control && e.KeyCode == Keys.A)
8:     {
9:         for (var i = 0; i < lbFiles.Items.Count; i++)
10:        {
11:            lbFiles.SetSelected(i, true);
12:        }
13:    }
14: }

```

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 =
9:                 MessageBox.Show(
10:                     resource.GetString("DeleteWarning1") + lbFiles.SelectedItems.Count +
11:                     resource.GetString("DeleteWarning2"),
12:                     resource.GetString("AppTitle"),
13:                     MessageBoxButtons.YesNo, MessageBoxIcon.Warning);
14:             if (deleteDialog == DialogResult.Yes)
15:                 DeleteFilesFromListBox();
16:         }
17:         else
18:             MessageBox.Show(resource.GetString("NoSelectedFile"),
19:                             resource.GetString("AppTitle"),
20:                             MessageBoxButtons.OK, MessageBoxIcon.Error);
21:     }
22:     else
23:         MessageBox.Show(resource.GetString("NoFile"), resource.GetString("AppTitle"),
24:                         MessageBoxButtons.OK, MessageBoxIcon.Error);
25: }

```

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"),
7:             MessageBoxButtons.OK,
8:             MessageBoxIcon.Error);
9: }
```

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"),
7:             MessageBoxButtons.OK,
8:             MessageBoxIcon.Error);
9: }
```

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,
8:             MessageBoxIcon.Error);
9: }
```

```
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,
8:             MessageBoxIcon.Error);
9: }
```

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 45)

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 28)	This is file AssemblyInfo.cs.
BaseException.cs (see page 29)	This is file BaseException.cs.
ExceptionExtensions.cs (see page 29)	This is file ExceptionExtensions.cs.
FrmMain.cs (see page 30)	This is file FrmMain.cs.
FrmMain.Designer.cs (see page 40)	This is file FrmMain.Designer.cs.
PdfCombiner.csproj (see page 44)	This is file PdfCombiner.csproj.
PdfCombiner.sln (see page 44)	This is file PdfCombiner.sln.
Program.cs (see page 45)	This is file Program.cs.
Resources.Designer.cs (see page 45)	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.
Settings.Designer.cs (see page 47)	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.

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.Globalization;
8: using System.IO;
9: using System.Linq;
10: using System.Reflection;
11: using System.Resources;
12: using System.Windows.Forms;
13:
14: namespace PdfCombiner
15: {
16:     public partial class FrmMain : Form
17:     {
18:         public static ResourceManager resource;
19:
20:         // Culture lists whose resources have been added to project
21:         private static readonly List<string> languageList = new List<string>
22:             {"EN", "TR", "DE", "FR", "RU", "ES", "IT", "ZH", "AR"};
23:
24:         public FrmMain()
25:         {
26:             InitializeComponent();
27:         }
28:
29:         #region Form Init, Language Settings and Finalize Methods
30:
31:         /// <summary>
32:         /// This function is used to terminate application
```



```

33:         /// </summary>
34:         /// <param name="sender">The sender info (For example Main Form)</param>
35:         /// <param name="e">Event Arguments</param>
36:         private void FrmMain_FormClosed(object sender, FormClosedEventArgs e)
37:         {
38:             MessageBox.Show(resource.GetString("ThanksMessage"),
resource.GetString("AppTitle"), MessageBoxButtons.OK,
39:             MessageBoxIcon.Information);
40:             Application.Exit();
41:         }
42:
43:         /// <summary>
44:         /// This function is used to assign menu to listbox
45:         /// When form is shown
46:         /// </summary>
47:         /// <param name="sender">The sender info (For example Main Form)</param>
48:         /// <param name="e">Event Arguments</param>
49:         private void FrmMain_Shown(object sender, EventArgs e)
50:         {
51:             lbFiles.ContextMenuStrip = menuStrip;
52:             cmbLanguage.DropDownStyle = ComboBoxStyle.DropDownList;
53:             cmbLanguage.Items.Clear();
54:             foreach (var item in languageList)
55:                 cmbLanguage.Items.Add(item);
56:             resource = new ResourceManager("PdfCombiner.Resources.AppResources-tr",
Assembly.GetExecutingAssembly());
57:             FormMembersNameInitialization();
58:         }
59:
60:         /// <summary>
61:         /// This function is used to set the resource file
62:         /// And with this, you can see form elements, info messages etc. with the
selected
63:         /// resource file contents. That is used to multi language support
64:         /// </summary>
65:         /// <param name="sender">The sender info (For example Main Form)</param>
66:         /// <param name="e">Event Arguments</param>
67:         private void cmbLanguage_SelectedIndexChanged(object sender, EventArgs e)
68:         {
69:             if (cmbLanguage.SelectedIndex != -1)
70:             {
71:                 try
72:                 {
73:                     resource = new ResourceManager(
"PdfCombiner.Resources.AppResources-" +
74:                     cmbLanguage.SelectedItem.ToString().ToLower(new
75:                     CultureInfo("en-US")),
Assembly.GetExecutingAssembly());
76:                     FormMembersNameInitialization();
77:                 }
78:                 catch (Exception)
79:                 {
80:                     resource = new
81:                     ResourceManager("PdfCombiner.Resources.AppResources-en",
Assembly.GetExecutingAssembly());
82:                     FormMembersNameInitialization();
83:                 }
84:             }
85:         }
86:     }
87:
88:     /// <summary>
89:     /// This method is used to change text of form members by the
90:     /// Resource file which is selected
91:     /// </summary>
92:     private void FormMembersNameInitialization()
93:     {
94:         ActiveForm.Text = resource.GetString("AppTitle");
95:         btnAddFile.Text = resource.GetString("AddFile");
96:         btnAddFolder.Text = resource.GetString("AddFolder");

```

```

97:         btnClearList.Text = resource.GetString("ClearFileList");
98:         btnCombineITextSharp.Text = resource.GetString("CombineFilesITextSharp");
99:         btnCombinePdfSharp.Text = resource.GetString("CombineFilesPdfSharp");
100:        menuItemDelete.Text = resource.GetString("Delete");
101:        menuItemOrderByPathAscending.Text =
resource.GetString("OrderByPathAscending");
102:        menuItemOrderByPathDescending.Text =
resource.GetString("OrderByPathDescending");
103:        menuItemOrderByNameAscending.Text =
resource.GetString("OrderByNameAscending");
104:        menuItemOrderByNameDescending.Text =
resource.GetString("OrderByNameDescending");
105:        cmbLanguage.Text = resource.GetString("SelectLanguage");
106:    }
107:
108:    #endregion
109:
110:    #region Add Item Methods
111:
112:    /// <summary>
113:    /// It is used to add single or multiple PDF files to combine
114:    /// When you choose file or files
115:    /// The listbox in the form will be filled with the name of the files
116:    /// Which you choose to combine
117:    /// </summary>
118:    /// <param name="sender">The sender info (For example Main Form)</param>
119:    /// <param name="e">Event Arguments</param>
120:    private void btnAddFile_Click(object sender, EventArgs e)
121:    {
122:        using (var dialogAddFile = new OpenFileDialog())
123:        {
124:            InitializeFileDialog(dialogAddFile);
125:            var result = dialogAddFile.ShowDialog();
126:            var addedFileCount = 0;
127:            if (result == DialogResult.OK && dialogAddFile.FileNames != null)
128:            {
129:                foreach (var file in dialogAddFile.FileNames)
130:                {
131:                    if (!lbFiles.Items.Contains(file))
132:                    {
133:                        lbFiles.Items.Add(file);
134:                        addedFileCount++;
135:                    }
136:                }
137:
138:                MessageBox.Show(addedFileCount +
resource.GetString("FileAddMessage"),
139:                    resource.GetString("AppTitle"), MessageBoxButtons.OK,
MessageBoxIcon.Information);
140:            }
141:        }
142:    }
143:
144:    /// <summary>
145:    /// This function is used to set options of file dialog like
146:    /// Filter, start path etc.
147:    /// </summary>
148:    /// <param name="dialogAddFile">File Dialog</param>
149:    private static void InitializeFileDialog(OpenFileDialog dialogAddFile)
150:    {
151:        dialogAddFile.Multiselect = true;
152:        dialogAddFile.Filter = resource.GetString("PdfFiles") + " (*.pdf)|*.pdf";
153:        dialogAddFile.InitialDirectory = Application.StartupPath;
154:        dialogAddFile.Title = resource.GetString("SelectPdfFile");
155:        dialogAddFile.DefaultExt = "PDF";
156:    }
157:
158:    /// <summary>
159:    /// This function is used to add PDF files in a folder which you choose in

```

```

160:         /// Folder Dialog recursively
161:         /// When you use it the added files will be seen in listbox
162:         /// </summary>
163:         /// <param name="sender">The sender info (For example Main Form)</param>
164:         /// <param name="e">Event Arguments</param>
165:         private void btnAddFolder_Click(object sender, EventArgs e)
166:         {
167:             using (var dialogAddFolder = new FolderBrowserDialog())
168:             {
169:                 var result = dialogAddFolder.ShowDialog();
170:                 if (result == DialogResult.OK &&
171:                     !string.IsNullOrEmpty(dialogAddFolder.SelectedPath))
172:                 {
173:                     var fileNames = Directory.GetFiles(dialogAddFolder.SelectedPath,
174:                         "*.pdf",
175:                         SearchOption.AllDirectories);
176:                     var addedFileCount = 0;
177:                     foreach (var file in fileNames)
178:                     {
179:                         if (!lbFiles.Items.Contains(file))
180:                         {
181:                             lbFiles.Items.Add(file);
182:                             addedFileCount++;
183:                         }
184:                     }
185:                     MessageBox.Show(addedFileCount +
186:                         resource.GetString("FileAddMessage"),
187:                         resource.GetString("AppTitle"), MessageBoxButtons.OK,
188:                         MessageBoxIcon.Information);
189:                 }
190:             }
191:         }
192:         #endregion
193:         #region Combine PDF Methods
194:         /// <summary>
195:         /// This function is used to take list of PDF files in listbox,
196:         /// Combine them with a single file in location which you choose in folder
197:         /// dialog
198:         /// And you can see the progress in progress bar
199:         /// It uses PdfSharp Nuget Package
200:         /// </summary>
201:         /// <param name="sender">The sender info (For example Main Form)</param>
202:         /// <param name="e">Event Arguments</param>
203:         private void btnCombinePdfSharp_Click(object sender, EventArgs e)
204:         {
205:             try
206:             {
207:                 if (lbFiles.Items.Count < 2)
208:                 {
209:                     MessageBox.Show(resource.GetString("CombineWarning"),
210:                         resource.GetString("AppTitle"),
211:                         MessageBoxButtons.OK, MessageBoxIcon.Error);
212:                 }
213:                 else
214:                 {
215:                     using (var dialogExport = new FolderBrowserDialog())
216:                     {
217:                         var result = dialogExport.ShowDialog();
218:                         if (result == DialogResult.OK &&
219:                             !string.IsNullOrEmpty(dialogExport.SelectedPath))
220:                         {
221:                             var outputFileName = dialogExport.SelectedPath + "/" +
222:                                 Guid.NewGuid() + ".pdf";
223:                             using (var outputFile = new PdfDocument())
224:                             {
225:                                 outputFile.Options.CompressContentStreams = true;
226:                             }
227:                         }
228:                     }
229:                 }
230:             }
231:             catch { }
232:         }
233:     }
234: }

```

```

outputFile.Options.EnableCcittCompressionForBilevelImages = true;
221:         outputFile.Options.FlateEncodeMode =
PdfFlateEncodeMode.BestCompression;
222:         var fileCount = lbFiles.Items.Count;
223:         var combinedFiles = 0;
224:
225:         for (int i = 0; i < lbFiles.Items.Count; i++)
226:         {
227:             try
228:             {
229:                 var inputDocument =
PdfReader.Open(lbFiles.Items[i].ToString(),
230:                 PdfDocumentOpenMode.Import);
231:                 var count = inputDocument.PageCount;
232:                 for (int idx = 0; idx < count; idx++)
233:                 {
234:                     var page = inputDocument.Pages[idx];
235:                     outputFile.AddPage(page);
236:                 }
237:
238:                 inputDocument.Close();
239:                 combinedFiles++;
240:             }
241:             catch (Exception)
242:             {
243:                 fileCount--;
244:             }
245:
246:             pbFiles.Value = (combinedFiles * 100 / fileCount);
247:             if (pbFiles.Value > pbFiles.Maximum)
248:                 pbFiles.Value = pbFiles.Maximum;
249:             ActiveForm.Text = "%" + pbFiles.Value;
250:         }
251:
252:         outputFile.Save(outputFileName);
253:         outputFile.Close();
254:
255:         MessageBox.Show(fileCount +
resource.GetString("CombineFileMessage") + outputFileName,
256:         resource.GetString("AppTitle"),
MessageBoxButtons.OK, MessageBoxIcon.Information);
257:     }
258:
259:     pbFiles.Value = pbFiles.Minimum;
260:     ActiveForm.Text = resource.GetString("AppTitle");
261: }
262: }
263: }
264: }
265: catch (Exception ex)
266: {
267:     MessageBox.Show(resource.GetString("CombineErrorMessage") +
ex.GetAllMessages(),
268:     resource.GetString("AppTitle"), MessageBoxButtons.OK,
MessageBoxIcon.Error);
269:     pbFiles.Value = pbFiles.Minimum;
270:     ActiveForm.Text = resource.GetString("AppTitle");
271: }
272: }
273:
274: /// <summary>
275: /// This function is used to take list of PDF files in listbox,
276: /// Combine them with a single file in location which you choose in folder
dialog
277: /// And you can see the progress in progress bar
278: /// It uses iTextSharp Nuget Package
279: /// </summary>
280: /// <param name="sender">The sender info (For example Main Form)</param>
281: /// <param name="e">Event Arguments</param>

```

```

282:         private void btnCombineITextSharp_Click(object sender, EventArgs e)
283:         {
284:             try
285:             {
286:                 if (lbFiles.Items.Count < 2)
287:                     MessageBox.Show(resource.GetString("CombineWarning"),
resource.GetString("AppTitle"),
288:                         MessageBoxButtons.OK, MessageBoxIcon.Error);
289:                 else
290:                 {
291:                     using (var dialogExport = new FolderBrowserDialog())
292:                     {
293:                         var result = dialogExport.ShowDialog();
294:                         if (result == DialogResult.OK &&
!string.IsNullOrEmpty(dialogExport.SelectedPath))
295:                         {
296:                             var outputFileName = dialogExport.SelectedPath + "/" +
Guid.NewGuid() + ".pdf";
297:                             var fileCount = lbFiles.Items.Count;
298:                             var combinedFiles = 0;
299:
300:                             var outputFile = new Document();
301:                             using (FileStream outputFileStream = new
FileStream(outputFileName, FileMode.Create))
302:                             {
303:                                 using (var pdfWriter = new
iTextSharp.text.pdf.PdfCopy(outputFile, outputFileStream))
304:                                 {
305:                                     if (pdfWriter == null)
306:                                     {
307:                                         return;
308:                                     }
309:
310:                                     pdfWriter.SetFullCompression();
311:                                     outputFile.Open();
312:                                     for (int i = 0; i < lbFiles.Items.Count; i++)
313:                                     {
314:                                         try
315:                                         {
316:                                             var pdfReader =
new
iTextSharp.text.pdf.PdfReader(lbFiles.Items[i].ToString());
317:                                             pdfReader.ConsolidateNamedDestinations();
318:                                             for (int j = 1; j <=
pdfReader.NumberOfPages; j++)
319:                                             {
320:                                                 var page =
pdfWriter.GetImportedPage(pdfReader, j);
321:                                                 pdfWriter.AddPage(page);
322:                                             }
323:                                             pdfReader.Close();
324:                                             combinedFiles++;
325:                                         }
326:                                         catch (Exception)
327:                                         {
328:                                             fileCount--;
329:                                         }
330:                                     }
331:
332:                                     pbFiles.Value = (combinedFiles * 100 /
fileCount);
333:
334:                                     if (pbFiles.Value > pbFiles.Maximum)
335:                                         pbFiles.Value = pbFiles.Maximum;
336:                                     ActiveForm.Text = "%" + pbFiles.Value;
337:                                 }
338:
339:                                 pdfWriter.Close();
340:                             }
341:

```

```

342:                outputFile.Close();
343:            }
344:
345:            MessageBox.Show(fileCount +
resource.GetString("CombineFileMessage") + outputFileName,
346:                resource.GetString("AppTitle"), MessageBoxButtons.OK,
MessageBoxIcon.Information);
347:
348:            pbFiles.Value = pbFiles.Minimum;
349:            ActiveForm.Text = resource.GetString("AppTitle");
350:        }
351:    }
352:
353:    }
354:    catch (Exception ex)
355:    {
356:        MessageBox.Show(resource.GetString("CombineErrorMessage") +
ex.GetAllMessages(),
357:            resource.GetString("AppTitle"), MessageBoxButtons.OK,
MessageBoxIcon.Error);
358:        pbFiles.Value = pbFiles.Minimum;
359:        ActiveForm.Text = resource.GetString("AppTitle");
360:    }
361:}
362:
363:    #endregion
364:
365:    #region Delete Item Methods
366:
367:    /// <summary>
368:    /// This function is used to operate some process in files in listbox which
are chosen
369:    /// </summary>
370:    /// <param name="sender">The sender info (For example Main Form)</param>
371:    /// <param name="e">Event Arguments</param>
372:    private void lbFiles_KeyDown(object sender, KeyEventArgs e)
373:    {
374:        // this is used to delete selected files in listbox when you press Delete
key
375:        if (e.KeyValue == Keys.Delete.GetHashCode())
376:            DeleteFilesFromListBox();
377:        // This is used to select all items in listbox when you press Ctrl + A key
combination
378:        if (e.Control && e.KeyCode == Keys.A)
379:        {
380:            for (var i = 0; i < lbFiles.Items.Count; i++)
381:            {
382:                lbFiles.SetSelected(i, true);
383:            }
384:        }
385:    }
386:
387:    /// <summary>
388:    /// This is used to delete selected items in listbox
389:    /// When menu item Delete is clicked and opened dialog
390:    /// </summary>
391:    /// <param name="sender">The sender info (For example Main Form)</param>
392:    /// <param name="e">Event Arguments</param>
393:    private void menuItemDelete_Click(object sender, EventArgs e)
394:    {
395:        if (lbFiles.Items.Count > 0)
396:        {
397:            if (lbFiles.SelectedItems.Count > 0)
398:            {
399:                var deleteDialog = new DialogResult();
400:                deleteDialog =
401:                    MessageBox.Show(
402:                        resource.GetString("DeleteWarning1") +
lbFiles.SelectedItems.Count +

```

```

403:                resource.GetString("DeleteWarning2"),
resource.GetString("AppTitle"),
404:                MessageBoxButtons.YesNo, MessageBoxIcon.Warning);
405:                if (deleteDialog == DialogResult.Yes)
406:                    DeleteFilesFromListBox();
407:            }
408:            else
409:                MessageBox.Show(resource.GetString("NoSelectedFile"),
resource.GetString("AppTitle"),
410:                MessageBoxButtons.OK, MessageBoxIcon.Error);
411:        }
412:        else
413:            MessageBox.Show(resource.GetString("NoFile"),
resource.GetString("AppTitle"), MessageBoxButtons.OK,
414:                MessageBoxIcon.Error);
415:    }
416:
417:    /// <summary>
418:    /// This method is used to Delete selected files from ListBox
419:    /// This is used different ways in this app
420:    /// So we created this as a method
421:    /// </summary>
422:    private void DeleteFilesFromListBox()
423:    {
424:        var allItems = new List<string>();
425:        var removedItems = new List<string>();
426:        var fileCount = lbFiles.SelectedItems.Count;
427:        for (int i = 0; i < lbFiles.SelectedItems.Count; i++)
428:        {
429:            removedItems.Add(lbFiles.SelectedItems[i].ToString());
430:        }
431:
432:        foreach (var item in lbFiles.Items)
433:        {
434:            if (!removedItems.Any(j => j == item.ToString()))
435:                allItems.Add(item.ToString());
436:        }
437:
438:        lbFiles.Items.Clear();
439:        foreach (var item in allItems)
440:        {
441:            lbFiles.Items.Add(item);
442:        }
443:
444:        MessageBox.Show(fileCount + resource.GetString("DeleteFileMessage"),
resource.GetString("AppTitle"),
445:            MessageBoxButtons.OK, MessageBoxIcon.Information);
446:    }
447:
448:    /// <summary>
449:    /// This function is used to clear the file list in listbox
450:    /// </summary>
451:    /// <param name="sender">The sender info (For example Main Form)</param>
452:    /// <param name="e">Event Arguments</param>
453:    private void btnClearList_Click(object sender, EventArgs e)
454:    {
455:        var fileCount = lbFiles.Items.Count;
456:        lbFiles.Items.Clear();
457:        MessageBox.Show(fileCount + resource.GetString("DeleteFileMessage"),
resource.GetString("AppTitle"),
458:            MessageBoxButtons.OK, MessageBoxIcon.Information);
459:    }
460:
461:    #endregion
462:
463:    #region Order Item Methods
464:
465:    /// <summary>
466:    /// This method is used to order elements in listbox ascending by full path

```



```

467:         /// </summary>
468:         /// <param name="sender">The sender info (For example Main Form)</param>
469:         /// <param name="e">Event Arguments</param>
470:         private void menuItemOrderByPathAscending_Click(object sender, EventArgs e)
471:         {
472:             if (lbFiles.Items.Count > 0)
473:                 lbFiles = SortItemsByPath(lbFiles, true);
474:             else
475:                 MessageBox.Show(resource.GetString("NoFile"),
resource.GetString("AppTitle"), MessageBoxButtons.OK,
476:                 MessageBoxIcon.Error);
477:         }
478:
479:         /// <summary>
480:         /// This method is used to order elements in listbox descending by full path
481:         /// </summary>
482:         /// <param name="sender">The sender info (For example Main Form)</param>
483:         /// <param name="e">Event Arguments</param>
484:         private void menuItemOrderByPathDescending_Click(object sender, EventArgs e)
485:         {
486:             if (lbFiles.Items.Count > 0)
487:                 lbFiles = SortItemsByPath(lbFiles, false);
488:             else
489:                 MessageBox.Show(resource.GetString("NoFile"),
resource.GetString("AppTitle"), MessageBoxButtons.OK,
490:                 MessageBoxIcon.Error);
491:         }
492:
493:         /// <summary>
494:         /// This method is used to order elements in listbox ascending by file name
495:         /// </summary>
496:         /// <param name="sender">The sender info (For example Main Form)</param>
497:         /// <param name="e">Event Arguments</param>
498:         private void menuItemOrderByNameAscending_Click(object sender, EventArgs e)
499:         {
500:             if (lbFiles.Items.Count > 0)
501:                 lbFiles = SortItemsByName(lbFiles, true);
502:             else
503:                 MessageBox.Show(resource.GetString("NoFile"),
resource.GetString("AppTitle"), MessageBoxButtons.OK,
504:                 MessageBoxIcon.Error);
505:         }
506:
507:         /// <summary>
508:         /// This method is used to order elements in listbox descending by file name
509:         /// </summary>
510:         /// <param name="sender">The sender info (For example Main Form)</param>
511:         /// <param name="e">Event Arguments</param>
512:         private void menuItemOrderByNameDescending_Click(object sender, EventArgs e)
513:         {
514:             if (lbFiles.Items.Count > 0)
515:                 lbFiles = SortItemsByName(lbFiles, false);
516:             else
517:                 MessageBox.Show(resource.GetString("NoFile"),
resource.GetString("AppTitle"), MessageBoxButtons.OK,
518:                 MessageBoxIcon.Error);
519:         }
520:
521:         /// <summary>
522:         /// This method is used to order items by full path in listbox
523:         /// And return them with the given order as parametre
524:         /// </summary>
525:         /// <param name="listBox">ListBox Info</param>
526:         /// <param name="ascending">Order Type is Ascending Or Not</param>
527:         private static ListBox SortItemsByPath(ListBox listBox, bool ascending)
528:         {
529:             List<object> items;
530:             items = listBox.Items.OfType<object>().ToList();
531:             listBox.Items.Clear();

```



```

532:         if (ascending)
533:             listBox.Items.AddRange(items.OrderBy(i => i).ToArray());
534:         else
535:             listBox.Items.AddRange(items.OrderByDescending(i => i).ToArray());
536:         return listBox;
537:     }
538:
539:     /// <summary>
540:     /// This method is used to order items by file name in listbox
541:     /// And return them with the given order as parametre
542:     /// </summary>
543:     /// <param name="listBox">ListBox Info</param>
544:     /// <param name="ascending">Order Type is Ascending Or Not</param>
545:     private static ListBox SortItemsByName(ListBox listBox, bool ascending)
546:     {
547:         var fileNameList = new List<string>();
548:         List<object> items;
549:         items = listBox.Items.OfType<object>().ToList();
550:         const char c = '\u005c';
551:         foreach (var item in items)
552:         {
553:             var fullPath = item.ToString();
554:             var list = fullPath.Split(c).ToList();
555:             if (list.Count > 0)
556:                 fileNameList.Add(list[list.Count - 1]);
557:         }
558:
559:         if (ascending)
560:             fileNameList = fileNameList.OrderBy(j => j).ToList();
561:         else
562:             fileNameList = fileNameList.OrderByDescending(j => j).ToList();
563:
564:         listBox.Items.Clear();
565:         foreach (var item in fileNameList)
566:             listBox.Items.Add(items.First(j => j.ToString().EndsWith(item)));
567:
568:         return listBox;
569:     }
570:
571: #endregion
572:
573: #region Item Drag and Drop Methods
574:
575:     /// <summary>
576:     /// This is used to control mouse actions and if that is not right click
577:     /// Start to drag and drop item in listbox
578:     /// </summary>
579:     /// <param name="sender">The sender info (For example Main Form)</param>
580:     /// <param name="e">Event Arguments</param>
581:     private void lbFiles_MouseDown(object sender, MouseEventArgs e)
582:     {
583:         if (!e.Button.Equals(MouseButtons.Right))
584:         {
585:             if (lbFiles.SelectedItem == null) return;
586:             lbFiles.DoDragDrop(lbFiles.SelectedItem, DragDropEffects.Move);
587:         }
588:     }
589:
590:     /// <summary>
591:     /// This is used to give effect while dragging and dropping an item in listbox
592:     /// </summary>
593:     /// <param name="sender">The sender info (For example Main Form)</param>
594:     /// <param name="e">Event Arguments</param>
595:     private void lbFiles_DragOver(object sender, DragEventArgs e)
596:     {
597:         e.Effect = DragDropEffects.Move;
598:     }
599:
600:     /// <summary>

```

```

601:         /// This is used to drag and drop an item in listbox
602:         /// Which is used to order books and combine them with that order
603:         /// </summary>
604:         /// <param name="sender">The sender info (For example Main Form)</param>
605:         /// <param name="e">Event Arguments</param>m>
606:         private void lbFiles_DragDrop(object sender, DragEventArgs e)
607:         {
608:             var point = lbFiles.PointToClient(new Point(e.X, e.Y));
609:             var index = this.lbFiles.IndexFromPoint(point);
610:             if (index < 0) index = this.lbFiles.Items.Count - 1;
611:             // type of string because file names stored in string in listbox
612:             var data = e.Data.GetData(typeof(string));
613:             this.lbFiles.Items.Remove(data);
614:             this.lbFiles.Items.Insert(index, data);
615:         }
616:
617:         #endregion
618:     }
619: }

```

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.AutoScaleMode.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 (see page 1)	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:
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Resources.Tools.StronglyTypedResourceBuilder",
"15.0.0.0")]
23:      [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
24:      [global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]
25:      internal class Resources {
26:
27:          private static global::System.Resources.ResourceManager resourceMan;
28:
29:          private static global::System.Globalization.CultureInfo resourceCulture;
30:
31:
[global::System.Diagnostics.CodeAnalysis.SuppressMessageAttribute("Microsoft.Performance",
"CA1811:AvoidUncalledPrivateCode")]
32:          internal Resources() {
33:          }
34:
35:          /// <summary>
36:          ///     Returns the cached ResourceManager instance used by this class.
37:          /// </summary>
38:
[global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableState.Advanced)]
39:          internal static global::System.Resources.ResourceManager ResourceManager {
40:              get {
41:                  if (object.ReferenceEquals(resourceMan, null)) {
42:                      global::System.Resources.ResourceManager temp = new
global::System.Resources.ResourceManager("PdfCombiner.Properties.Resources",
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:
[global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableState.Advanced)]
54:          internal static global::System.Globalization.CultureInfo Culture {
55:              get {
56:                  return resourceCulture;
57:              }
58:              set {
59:                  resourceCulture = value;
60:              }
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:     [global::System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.VisualStudio.Editors.SettingsDesigner.SettingsSingleFileGenerator",
16: "11.0.0.0")]
17:     internal sealed partial class Settings :
18: global::System.Configuration.ApplicationSettingsBase
19:     {
20:         private static Settings defaultInstance =
21: ((Settings)(global::System.Configuration.ApplicationSettingsBase.Synchronized(new
22: Settings())));
23:     }
```

Namespaces

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

Index

A

AssemblyInfo.cs 28

B

BaseException class 4

about BaseException class 4

BaseException 5

BaseException.cs 29

E

ExceptionExtensions class 5

about ExceptionExtensions class 5

GetAllMessages 6

ExceptionExtensions.cs 29

F

Files 28

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 16

components 9

DeleteFilesFromListBox 16

Dispose 17

FormMembersNameInitialization 17

FrmMain 8

FrmMain_FormClosed 18

FrmMain_Shown 18

InitializeComponent 18

InitializeFileDialog 22

languageList 9

lbFiles 10

lbFiles_DragDrop 22

lbFiles_DragOver 23

lbFiles_KeyDown 23

lbFiles_MouseDown 23

lbIDetails 10

menuItemDelete 10

menuItemDelete_Click 24

menuItemOrderByNameAscending 10

menuItemOrderByNameAscending_Click 24

menuItemOrderByNameDescending 10

menuItemOrderByNameDescending_Click 25

menuItemOrderByPathAscending 10

menuItemOrderByPathAscending_Click 25

menuItemOrderByPathDescending 11

menuItemOrderByPathDescending_Click 26

menuStrip 11

pbFiles 11

resource 11

SortItemsByName 26

SortItemsByPath 27

FrmMain.cs 30

FrmMain.Designer.cs 40

P

PdfCombiner 1

PdfCombiner namespace 1

Classes 4

PdfCombiner.BaseException 4

PdfCombiner.BaseException.BaseException 5

PdfCombiner.csproj 44

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.menuItemOrderByPathDescending_Click 26
PdfCombiner.FrmMain.btnAddFolder_Click 12	PdfCombiner.FrmMain.menuStrip 11
PdfCombiner.FrmMain.btnClearList 9	PdfCombiner.FrmMain.pbFiles 11
PdfCombiner.FrmMain.btnClearList_Click 12	PdfCombiner.FrmMain.resource 11
PdfCombiner.FrmMain.btnCombineTextSharp 9	PdfCombiner.FrmMain.SortItemsByName 26
PdfCombiner.FrmMain.btnCombineTextSharp_Click 13	PdfCombiner.FrmMain.SortItemsByPath 27
PdfCombiner.FrmMain.btnCombinePdfSharp 9	PdfCombiner.Program 27
PdfCombiner.FrmMain.btnCombinePdfSharp_Click 14	PdfCombiner.Program.Main 27
PdfCombiner.FrmMain.cmbLanguage 9	PdfCombiner.Properties 1
PdfCombiner.FrmMain.cmbLanguage_SelectedIndexChanged 16	PdfCombiner.Properties namespace 1
PdfCombiner.FrmMain.components 9	Classes 1
PdfCombiner.FrmMain.DeleteFilesFromListBox 16	PdfCombiner.Properties.Resources 1
PdfCombiner.FrmMain.Dispose 17	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 18	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 22	PdfCombiner.Properties.Settings 3
PdfCombiner.FrmMain.languageList 9	PdfCombiner.Properties.Settings.defaultInstance 4
PdfCombiner.FrmMain.lbFiles 10	PdfCombiner.sln 44
PdfCombiner.FrmMain.lbFiles_DragDrop 22	Program class 27
PdfCombiner.FrmMain.lbFiles_DragOver 23	about Program class 27
PdfCombiner.FrmMain.lbFiles_KeyDown 23	Main 27
PdfCombiner.FrmMain.lbFiles_MouseDown 23	Program.cs 45
PdfCombiner.FrmMain.lblDetails 10	
PdfCombiner.FrmMain.menuItemDelete 10	R
PdfCombiner.FrmMain.menuItemDelete_Click 24	Resources class 1
PdfCombiner.FrmMain.menuItemOrderByNameAscending 10	about Resources class 1
PdfCombiner.FrmMain.menuItemOrderByNameAscending_Click 24	AppTitle 3
PdfCombiner.FrmMain.menuItemOrderByNameDescending 10	Culture 3
PdfCombiner.FrmMain.menuItemOrderByNameDescending_Click 25	resourceCulture 2
PdfCombiner.FrmMain.menuItemOrderByPathAscending 10	resourceMan 3
PdfCombiner.FrmMain.menuItemOrderByPathAscending_Click 25	ResourceManager 3
PdfCombiner.FrmMain.menuItemOrderByPathDescending 11	Resources 2
	Resources.Designer.cs 45
	S
	Settings class 3

about Settings class 3

defaultInstance 4

Settings.Designer.cs 47