

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	5
ExceptionExtensions Class	5
ExceptionExtensions Methods	6
FrmMain Class	6
FrmMain.FrmMain Constructor	7
FrmMain Fields	8
FrmMain Methods	9
Program Class	17
Program Methods	17
Files	18
AssemblyInfo.cs	18
Exceptions.cs	19
FrmMain.cs	20
FrmMain.Designer.cs	25
PdfCombiner.csproj	27
PdfCombiner.sln	27
Program.cs	27
Resources.Designer.cs	28
Settings.Designer.cs	29
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 17)	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",
"4.0.0.0")]
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
[global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]
internal class Resources;
```


File

Resources.Designer.cs ([see page 28](#))



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
	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: {
4: }
```

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.

1.1.1.1.2.2 Resources.resourceMan Field

C#

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

Description

This is resourceMan, a member of class Resources.

1.1.1.1.3 Resources Properties

1.1.1.1.3.1 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;
```

1.1.1.1.3.2 Resources.ResourceManager Property

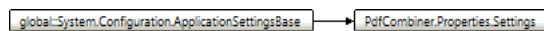
Returns the cached ResourceManager instance used by this class.

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 29](#))


Description

This is class PdfCombiner.Properties.Settings.

Settings Fields

	Name	Description
	defaultInstance (see page 4)	This is defaultInstance, a member of class Settings.

Settings Properties

	Name	Description
	Default (see page 4)	This is Default, 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.

1.1.1.1.2.2 Settings Properties

1.1.1.1.2.2.1 Settings.Default Property

C#

```
public static Settings Default;
```





Description

This is Default, a member of class Settings.

1.1.2 Classes

The following table lists classes in this documentation.

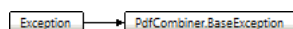
Classes

	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 17)	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

Exceptions.cs ([see page 19](#))

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

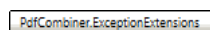
This is BaseException, a member of class BaseException.

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

Exceptions.cs ( see page 19)

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

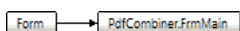
Parameters	Description
this Exception exception	Exception

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 25)











Description

This is class PdfCombiner.FrmMain.











Methods

	Name	Description
	FrmMain ( see page 7)	This is FrmMain, a member of class FrmMain.

FrmMain Fields

	Name	Description
	AppTitle (see page 8)	This is AppTitle, a member of class FrmMain.
	btnAddFile (see page 8)	This is btnAddFile, a member of class FrmMain.
	btnAddFolder (see page 8)	This is btnAddFolder, a member of class FrmMain.
	btnClearList (see page 8)	This is btnClearList, a member of class FrmMain.
	btnCombineTextSharp (see page 8)	This is btnCombineTextSharp, a member of class FrmMain.
	btnCombinePdfSharp (see page 8)	This is btnCombinePdfSharp, a member of class FrmMain.
	components (see page 8)	Required designer variable.
	lbFiles (see page 9)	This is lbFiles, a member of class FrmMain.
	lblDetails (see page 9)	This is lblDetails, a member of class FrmMain.
	pbFiles (see page 9)	This is pbFiles, a member of class FrmMain.

FrmMain Methods

	Name	Description
	btnAddFile_Click (see page 9)	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 10)	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 10)	This function is used to clear the file list in listbox
	btnCombineTextSharp_Click (see page 11)	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 12)	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
	Dispose (see page 13)	Clean up any resources being used.
	FrmMain_FormClosed (see page 14)	This function is used to terminate application
	InitializeComponent (see page 14)	Required method for Designer support - do not modify the contents of this method with the code editor.
	InitializeFileDialog (see page 16)	This function is used to set options of file dialog like Filter, start path etc.
	lbFiles_KeyDown (see page 16)	This function is used to delete files in listbox which are chosen If you press the Delete button in your keyboard The selected files will be deleted from the list

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.AppTitle Field

C#

```
public const string AppTitle = "PDF Combiner";
```

Description

This is AppTitle, a member of class FrmMain.

1.1.2.3.2.2 FrmMain.btnAddFile Field

C#

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

Description

This is btnAddFile, a member of class FrmMain.

1.1.2.3.2.3 FrmMain.btnAddFolder Field

C#

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

Description

This is btnAddFolder, a member of class FrmMain.

1.1.2.3.2.4 FrmMain.btnClearList Field

C#

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

Description

This is btnClearList, a member of class FrmMain.

1.1.2.3.2.5 FrmMain.btnCombineITextSharp Field

C#

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

Description

This is btnCombineITextSharp, a member of class FrmMain.

1.1.2.3.2.6 FrmMain.btnCombinePdfSharp Field

C#

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

Description

This is btnCombinePdfSharp, a member of class FrmMain.

1.1.2.3.2.7 FrmMain.components Field

Required designer variable.

C#

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

1.1.2.3.2.8 FrmMain.lbFiles Field**C#**

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

Description

This is lbFiles, a member of class FrmMain.

1.1.2.3.2.9 FrmMain.lblDetails Field**C#**

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

Description

This is lblDetails, a member of class FrmMain.

1.1.2.3.2.10 FrmMain.pbFiles Field**C#**

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

Description

This is pbFiles, a member of class FrmMain.

1.1.2.3.3 FrmMain Methods**1.1.2.3.3.1 FrmMain.btnAddFile_Click Method**

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

C#

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

Parameters

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

Body Source

```
1: private void btnAddFile_Click(object sender, EventArgs e)
2: {
3:     using (var dialogAddFile = new OpenFileDialog())
4:     {
5:         InitializeFileDialog(dialogAddFile);
6:         var result = dialogAddFile.ShowDialog();
7:         var addedFileCount = 0;
8:         if (result == DialogResult.OK && dialogAddFile.FileNames != null)
9:         {
10:             foreach (var file in dialogAddFile.FileNames)
11:             {
12:                 if (!lbFiles.Items.Contains(file))
13:                 {
```

```

14:                 lbFiles.Items.Add(file);
15:                 addedFileCount++;
16:             }
17:         }
18:         MessageBox.Show(addedFileCount + " file/s successfully added to the list",
AppTitle, MessageBoxButtons.OK, MessageBoxIcon.Information);
19:     }
20: }
21: }

```

1.1.2.3.3.2 FrmMain.btnAddFolder_Click Method

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

C#

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

Parameters

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

Body Source

```

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

```

1.1.2.3.3.3 FrmMain.btnClearList_Click Method

This function is used to clear the file list in listbox

C#

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

Parameters

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

Body Source

```
1: private void btnClearList_Click(object sender, EventArgs e)
```

```

2: {
3:     var fileCount = lbFiles.Items.Count;
4:     lbFiles.Items.Clear();
5:     MessageBox.Show(fileCount + " file/s successfully deleted from the list", AppTitle,
6:     MessageBoxButtons.OK, MessageBoxIcon.Information);
6: }

```

1.1.2.3.3.4 FrmMain.btnCombineITextSharp_Click Method

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

C#

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

Parameters

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

Body Source

```

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

```

```

39:                pdfWriter.AddPage(page);
40:            }
41:            pdfReader.Close();
42:            combinedFiles++;
43:        }
44:        catch (Exception)
45:        {
46:            fileCount--;
47:        }
48:
49:        pbFiles.Value = (combinedFiles * 100 / fileCount);
50:        if (pbFiles.Value > pbFiles.Maximum)
51:            pbFiles.Value = pbFiles.Maximum;
52:    }
53:    pdfWriter.Close();
54:    outputFile.Close();
55:    }
56:
57:    MessageBox.Show(fileCount + " PDF files successfully combined in "
+ outputFileName, AppTitle, MessageBoxButtons.OK, MessageBoxIcon.Information);
58:
59:    pbFiles.Value = pbFiles.Minimum;
60:    }
61:    }
62:    }
63:    }
64:    catch (Exception ex)
65:    {
66:        MessageBox.Show("There is an error while combining PDF files in list. Details:
" + ex.GetAllMessages(), AppTitle, MessageBoxButtons.OK, MessageBoxIcon.Error);
67:        pbFiles.Value = pbFiles.Minimum;
68:    }
69: }

```

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:         {
7:             MessageBox.Show("There must be at least 2 PDF files to combine", AppTitle,
MessageBoxButtons.OK, MessageBoxIcon.Error);
8:         }
9:         else
10:        {
11:            using (var dialogExport = new FolderBrowserDialog())
12:            {
13:                var result = dialogExport.ShowDialog();
14:                if (result == DialogResult.OK &&
!string.IsNullOrEmpty(dialogExport.SelectedPath))
15:                {

```

```

16:         var outputFileName = dialogExport.SelectedPath + "/" +
Guid.NewGuid().ToString() + ".pdf";
17:         var outputFile = new PdfDocument();
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(), PdfDocumentOpenMode.Import);
29:                 var count = inputDocument.PageCount;
30:                 for (int idx = 0; idx < count; idx++)
31:                 {
32:                     var page = inputDocument.Pages[idx];
33:                     outputFile.AddPage(page);
34:                 }
35:                 inputDocument.Close();
36:                 combinedFiles++;
37:             }
38:             catch (Exception)
39:             {
40:                 fileCount--;
41:             }
42:
43:             pbFiles.Value = (combinedFiles * 100 / fileCount);
44:             if (pbFiles.Value > pbFiles.Maximum)
45:                 pbFiles.Value = pbFiles.Maximum;
46:         }
47:         outputFile.Save(outputFileName);
48:         outputFile.Close();
49:
50:         MessageBox.Show(fileCount + " PDF files successfully combined in "
+ outputFileName, AppTitle, MessageBoxButtons.OK, MessageBoxIcon.Information);
51:
52:         pbFiles.Value = pbFiles.Minimum;
53:     }
54: }
55: }
56: }
57: catch (Exception ex)
58: {
59:     MessageBox.Show("There is an error while combining PDF files in list. Details:
" + ex.GetAllMessages(), AppTitle, MessageBoxButtons.OK, MessageBoxIcon.Error);
60:     pbFiles.Value = pbFiles.Minimum;
61: }
62: }

```

1.1.2.3.3.6 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: }

```

1.1.2.3.3.7 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("Thanks for using this app", AppTitle, MessageBoxButtons.OK,
4:     MessageBoxIcon.Information);
5:     Application.Exit();
6: }

```

1.1.2.3.3.8 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.btnCombinePdfSharp = new System.Windows.Forms.Button();
4:     this.btnAddFolder = new System.Windows.Forms.Button();
5:     this.btnAddFile = new System.Windows.Forms.Button();
6:     this.btnClearList = new System.Windows.Forms.Button();
7:     this.lbFiles = new System.Windows.Forms.ListBox();
8:     this.lblDetails = new System.Windows.Forms.Label();
9:     this.pbFiles = new System.Windows.Forms.ProgressBar();
10:    this.btnCombineITextSharp = new System.Windows.Forms.Button();
11:    this.SuspendLayout();
12:    //
13:    // btnCombinePdfSharp
14:    //
15:    this.btnCombinePdfSharp.Location = new System.Drawing.Point(487, 12);
16:    this.btnCombinePdfSharp.Name = "btnCombinePdfSharp";
17:    this.btnCombinePdfSharp.Size = new System.Drawing.Size(169, 23);
18:    this.btnCombinePdfSharp.TabIndex = 0;
19:    this.btnCombinePdfSharp.Text = "Combine Files (PdfSharp)";
20:    this.btnCombinePdfSharp.UseVisualStyleBackColor = true;
21:    this.btnCombinePdfSharp.Click += new
22:    System.EventHandler(this.btnCombinePdfSharp_Click);
23:    //
24:    // btnAddFolder
25:    //

```



```

25:     this.btnAddFolder.Location = new System.Drawing.Point(12, 38);
26:     this.btnAddFolder.Name = "btnAddFolder";
27:     this.btnAddFolder.Size = new System.Drawing.Size(156, 23);
28:     this.btnAddFolder.TabIndex = 2;
29:     this.btnAddFolder.Text = "Add Folder";
30:     this.btnAddFolder.UseVisualStyleBackColor = true;
31:     this.btnAddFolder.Click += new System.EventHandler(this.btnAddFolder_Click);
32:     //
33:     // btnAddFile
34:     //
35:     this.btnAddFile.Location = new System.Drawing.Point(12, 12);
36:     this.btnAddFile.Name = "btnAddFile";
37:     this.btnAddFile.Size = new System.Drawing.Size(156, 23);
38:     this.btnAddFile.TabIndex = 3;
39:     this.btnAddFile.Text = "Add Files";
40:     this.btnAddFile.UseVisualStyleBackColor = true;
41:     this.btnAddFile.Click += new System.EventHandler(this.btnAddFile_Click);
42:     //
43:     // btnClearList
44:     //
45:     this.btnClearList.Location = new System.Drawing.Point(174, 12);
46:     this.btnClearList.Name = "btnClearList";
47:     this.btnClearList.Size = new System.Drawing.Size(90, 49);
48:     this.btnClearList.TabIndex = 4;
49:     this.btnClearList.Text = "Clear File List";
50:     this.btnClearList.UseVisualStyleBackColor = true;
51:     this.btnClearList.Click += new System.EventHandler(this.btnClearList_Click);
52:     //
53:     // lbFiles
54:     //
55:     this.lbFiles.FormattingEnabled = true;
56:     this.lbFiles.Location = new System.Drawing.Point(12, 88);
57:     this.lbFiles.Name = "lbFiles";
58:     this.lbFiles.SelectionMode = System.Windows.Forms.SelectionMode.MultiExtended;
59:     this.lbFiles.Size = new System.Drawing.Size(644, 186);
60:     this.lbFiles.TabIndex = 5;
61:     this.lbFiles.KeyDown += new
System.Windows.Forms.KeyEventHandler(this.lbFiles_KeyDown);
62:     //
63:     // lblDetails
64:     //
65:     this.lblDetails.AutoSize = true;
66:     this.lblDetails.Location = new System.Drawing.Point(12, 65);
67:     this.lblDetails.Name = "lblDetails";
68:     this.lblDetails.Size = new System.Drawing.Size(0, 13);
69:     this.lblDetails.TabIndex = 6;
70:     //
71:     // pbFiles
72:     //
73:     this.pbFiles.Location = new System.Drawing.Point(270, 12);
74:     this.pbFiles.Name = "pbFiles";
75:     this.pbFiles.Size = new System.Drawing.Size(211, 49);
76:     this.pbFiles.TabIndex = 7;
77:     //
78:     // btnCombineITextSharp
79:     //
80:     this.btnCombineITextSharp.Location = new System.Drawing.Point(487, 38);
81:     this.btnCombineITextSharp.Name = "btnCombineITextSharp";
82:     this.btnCombineITextSharp.Size = new System.Drawing.Size(169, 23);
83:     this.btnCombineITextSharp.TabIndex = 9;
84:     this.btnCombineITextSharp.Text = "Combine Files (iTextSharp)";
85:     this.btnCombineITextSharp.UseVisualStyleBackColor = true;
86:     this.btnCombineITextSharp.Click += new
System.EventHandler(this.btnCombineITextSharp_Click);
87:     //
88:     // FrmMain
89:     //
90:     this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
91:     this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

```

```

92:     this.ClientSize = new System.Drawing.Size(668, 275);
93:     this.Controls.Add(this.btnCombineITextSharp);
94:     this.Controls.Add(this.pbFiles);
95:     this.Controls.Add(this.lblDetails);
96:     this.Controls.Add(this.lbFiles);
97:     this.Controls.Add(this.btnClearList);
98:     this.Controls.Add(this.btnAddFile);
99:     this.Controls.Add(this.btnAddFolder);
100:    this.Controls.Add(this.btnCombinePdfSharp);
101:    this.Name = "FrmMain";
102:    this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
103:    this.Text = "PDF Combiner";
104:    this.FormClosed += new
System.Windows.Forms.FormClosedEventHandler(this.FrmMain_FormClosed);
105:    this.ResumeLayout(false);
106:    this.PerformLayout();
107:
108: }

```

1.1.2.3.3.9 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 = "pdf files (*.pdf)|*.pdf";
5:     dialogAddFile.InitialDirectory = Application.StartupPath;
6:     dialogAddFile.Title = "Select PDF File/s";
7:     dialogAddFile.DefaultExt = "pdf";
8: }

```

1.1.2.3.3.10 FrmMain.lbFiles_KeyDown Method

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

C#

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

Parameters

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

Body Source

```

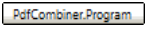
1: private void lbFiles_KeyDown(object sender, KeyEventArgs e)
2: {
3:     if (e.KeyValue == Keys.Delete.GetHashCode())
4:     {
5:         var allItems = new List<string>();
6:         var removedItems = new List<string>();
7:         var fileCount = lbFiles.SelectedItems.Count;
8:         for (int i = 0; i < lbFiles.SelectedItems.Count; i++)
9:         {

```

```
10:         removedItems.Add(lbFiles.SelectedItems[i].ToString());
11:     }
12:     foreach (var item in lbFiles.Items)
13:     {
14:         if (!removedItems.Any(j => j == item.ToString()))
15:             allItems.Add(item.ToString());
16:     }
17:     lbFiles.Items.Clear();
18:     foreach (var item in allItems)
19:     {
20:         lbFiles.Items.Add(item);
21:     }
22:     MessageBox.Show(fileCount + " file/s are deleted from the list.", AppTitle,
23:         MessageBoxButtons.OK, MessageBoxIcon.Information);
24: }
```

1.1.2.4 Program Class

Class Hierarchy



C#

```
public static class Program;
```

File

Program.cs (📄 see page 27)

Description

This is class PdfCombiner.Program.

Program Methods

	Name	Description
🔧📄	Main (📄 see page 17)	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 18)	This is file AssemblyInfo.cs.
Exceptions.cs (see page 19)	This is file Exceptions.cs.
FrmMain.cs (see page 20)	This is file FrmMain.cs.
FrmMain.Designer.cs (see page 25)	This is file FrmMain.Designer.cs.
PdfCombiner.csproj (see page 27)	This is file PdfCombiner.csproj.
PdfCombiner.sln (see page 27)	This is file PdfCombiner.sln.
Program.cs (see page 27)	This is file Program.cs.
Resources.Designer.cs (see page 28)	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 29)	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 Exceptions.cs

This is file Exceptions.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:
25:        public static class ExceptionExtensions
26:        {
27:            /// <summary>
28:            /// This method is used to take all messages with stack trace info in exception
29:            /// </summary>
30:            /// <param name="exception">Exception</param>
31:            /// <returns>Detailed Error Message</returns>
32:            public static string GetAllMessages(this Exception exception)
33:            {
34:                var messages = exception.Message;
35:                if (exception.InnerException != null)
36:                    messages += " " + exception.InnerException.GetAllMessages();
37:                if (!string.IsNullOrEmpty(exception.StackTrace))
38:                    messages += " " + exception.StackTrace;
39:                if (!string.IsNullOrEmpty(exception.Source))
40:                    messages += " " + exception.Source;
41:                return messages;
42:            }
43:        }
44:    }

```

Namespaces

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

1.2.3 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.IO;
7: using System.Linq;
8: using System.Windows.Forms;
9:
10: namespace PdfCombiner
11: {
12:     public partial class FrmMain : Form
13:     {
14:         public const string AppTitle = "PDF Combiner";
15:
16:         public FrmMain()
17:         {
18:             InitializeComponent();
19:         }
20:
21:         /// <summary>
22:         /// This function is used to terminate application
23:         /// </summary>
24:         /// <param name="sender">The sender info (For example Main Form)</param>
25:         /// <param name="e">Event Arguments</param>
26:         private void FrmMain_FormClosed(object sender, FormClosedEventArgs e)
27:         {
28:             MessageBox.Show("Thanks for using this app", AppTitle,
29:             MessageBoxButtons.OK, MessageBoxIcon.Information);
30:             Application.Exit();
31:         }
32:
33:         /// <summary>
34:         /// It is used to add single or multiple PDF files to combine
35:         /// When you choose file or files
36:         /// The listbox in the form will be filled with the name of the files
37:         /// Which you choose to combine
38:         /// </summary>
39:         /// <param name="sender">The sender info (For example Main Form)</param>
40:         /// <param name="e">Event Arguments</param>
41:         private void btnAddFile_Click(object sender, EventArgs e)
42:         {
43:             using (var dialogAddFile = new OpenFileDialog())
44:             {
45:                 InitializeFileDialog(dialogAddFile);
46:                 var result = dialogAddFile.ShowDialog();
47:                 var addedFileCount = 0;
48:                 if (result == DialogResult.OK && dialogAddFile.FileNames != null)
49:                 {
50:                     foreach (var file in dialogAddFile.FileNames)
51:                     {
52:                         if (!lbFiles.Items.Contains(file))
53:                         {
54:                             lbFiles.Items.Add(file);
55:                             addedFileCount++;
56:                         }
57:                     }
58:                     MessageBox.Show(addedFileCount + " file/s successfully added to
59: the list", AppTitle, MessageBoxButtons.OK, MessageBoxIcon.Information);

```

```

58:         }
59:     }
60: }
61:
62: /// <summary>
63: /// This function is used to set options of file dialog like
64: /// Filter, start path etc.
65: /// </summary>
66: /// <param name="dialogAddFile">File Dialog</param>
67: private static void InitializeFileDialog(OpenFileDialog dialogAddFile)
68: {
69:     dialogAddFile.Multiselect = true;
70:     dialogAddFile.Filter = "pdf files (*.pdf)|*.pdf";
71:     dialogAddFile.InitialDirectory = Application.StartupPath;
72:     dialogAddFile.Title = "Select PDF File/s";
73:     dialogAddFile.DefaultExt = "pdf";
74: }
75:
76: /// <summary>
77: /// This function is used to clear the file list in listbox
78: /// </summary>
79: /// <param name="sender">The sender info (For example Main Form)</param>
80: /// <param name="e">Event Arguments</param>
81: private void btnClearList_Click(object sender, EventArgs e)
82: {
83:     var fileCount = lbFiles.Items.Count;
84:     lbFiles.Items.Clear();
85:     MessageBox.Show(fileCount + " file/s successfully deleted from the list",
AppTitle, MessageBoxButtons.OK, MessageBoxIcon.Information);
86: }
87:
88: /// <summary>
89: /// This function is used to add PDF files in a folder which you choose in
90: /// Folder Dialog recursively
91: /// When you use it the added files will be seen in listbox
92: /// </summary>
93: /// <param name="sender">The sender info (For example Main Form)</param>
94: /// <param name="e">Event Arguments</param>
95: private void btnAddFolder_Click(object sender, EventArgs e)
96: {
97:     using (var dialogAddFolder = new FolderBrowserDialog())
98:     {
99:         var result = dialogAddFolder.ShowDialog();
100:         if (result == DialogResult.OK &&
!string.IsNullOrEmpty(dialogAddFolder.SelectedPath))
101:         {
102:             var fileNames = Directory.GetFiles(dialogAddFolder.SelectedPath,
"*.pdf", SearchOption.AllDirectories);
103:             var addedFileCount = 0;
104:             foreach (var file in fileNames)
105:             {
106:                 if (!lbFiles.Items.Contains(file))
107:                 {
108:                     lbFiles.Items.Add(file);
109:                     addedFileCount++;
110:                 }
111:             }
112:             MessageBox.Show(addedFileCount + " file/s successfully added to
the list", AppTitle, MessageBoxButtons.OK, MessageBoxIcon.Information);
113:         }
114:     }
115: }
116:
117: /// <summary>
118: /// This function is used to delete files in listbox which are chosen
119: /// If you press the Delete button in your keyboard
120: /// The selected files will be deleted from the list
121: /// </summary>
122: /// <param name="sender">The sender info (For example Main Form)</param>

```

```

123:         /// <param name="e">Event Arguments</param>
124:         private void lbFiles_KeyDown(object sender, KeyEventArgs e)
125:         {
126:             if (e.KeyValue == Keys.Delete.GetHashCode())
127:             {
128:                 var allItems = new List<string>();
129:                 var removedItems = new List<string>();
130:                 var fileCount = lbFiles.SelectedItems.Count;
131:                 for (int i = 0; i < lbFiles.SelectedItems.Count; i++)
132:                 {
133:                     removedItems.Add(lbFiles.SelectedItems[i].ToString());
134:                 }
135:                 foreach (var item in lbFiles.Items)
136:                 {
137:                     if (!removedItems.Any(j => j == item.ToString()))
138:                         allItems.Add(item.ToString());
139:                 }
140:                 lbFiles.Items.Clear();
141:                 foreach (var item in allItems)
142:                 {
143:                     lbFiles.Items.Add(item);
144:                 }
145:                 MessageBox.Show(fileCount + " file/s are deleted from the list.",
AppTitle, MessageBoxButtons.OK, MessageBoxIcon.Information);
146:             }
147:         }
148:
149:         /// <summary>
150:         /// This function is used to take list of PDF files in listbox,
151:         /// Combine them with a single file in location which you choose in folder
dialog
152:         /// And you can see the progress in progress bar
153:         /// It uses PdfSharp Nuget Package
154:         /// </summary>
155:         /// <param name="sender">The sender info (For example Main Form)</param>
156:         /// <param name="e">Event Arguments</param>
157:         private void btnCombinePdfSharp_Click(object sender, EventArgs e)
158:         {
159:             try
160:             {
161:                 if (lbFiles.Items.Count < 2)
162:                 {
163:                     MessageBox.Show("There must be at least 2 PDF files to combine",
AppTitle, MessageBoxButtons.OK, MessageBoxIcon.Error);
164:                 }
165:                 else
166:                 {
167:                     using (var dialogExport = new FolderBrowserDialog())
168:                     {
169:                         var result = dialogExport.ShowDialog();
170:                         if (result == DialogResult.OK &&
!string.IsNullOrEmpty(dialogExport.SelectedPath))
171:                         {
172:                             var outputFileName = dialogExport.SelectedPath + "/" +
Guid.NewGuid().ToString() + ".pdf";
173:                             var outputFile = new PdfDocument();
174:                             outputFile.Options.CompressContentStreams = true;
175:                             outputFile.Options.EnableCcittCompressionForBilevelImages
= true;
176:                             outputFile.Options.FlateEncodeMode =
PdfFlateEncodeMode.BestCompression;
177:                             var fileCount = lbFiles.Items.Count;
178:                             var combinedFiles = 0;
179:
180:                             for (int i = 0; i < lbFiles.Items.Count; i++)
181:                             {
182:                                 try
183:                                 {
184:                                     var inputDocument =

```



```

PdfReader.Open(lbFiles.Items[i].ToString(), PdfDocumentOpenMode.Import);
185:         var count = inputDocument.PageCount;
186:         for (int idx = 0; idx < count; idx++)
187:         {
188:             var page = inputDocument.Pages[idx];
189:             outputFile.AddPage(page);
190:         }
191:         inputDocument.Close();
192:         combinedFiles++;
193:     }
194:     catch (Exception)
195:     {
196:         fileCount--;
197:     }
198:
199:     pbFiles.Value = (combinedFiles * 100 / fileCount);
200:     if (pbFiles.Value > pbFiles.Maximum)
201:         pbFiles.Value = pbFiles.Maximum;
202:     }
203:     outputFile.Save(outputFileName);
204:     outputFile.Close();
205:
206:     MessageBox.Show(fileCount + " PDF files successfully
combined in " + outputFileName, AppTitle, MessageBoxButtons.OK, MessageBoxIcon.Information);
207:
208:     pbFiles.Value = pbFiles.Minimum;
209: }
210: }
211: }
212: }
213: catch (Exception ex)
214: {
215:     MessageBox.Show("There is an error while combining PDF files in list.
Details: " + ex.GetAllMessages(), AppTitle, MessageBoxButtons.OK, MessageBoxIcon.Error);
216:     pbFiles.Value = pbFiles.Minimum;
217: }
218: }
219:
220: /// <summary>
221: /// This function is used to take list of PDF files in listbox,
222: /// Combine them with a single file in location which you choose in folder
dialog
223: /// And you can see the progress in progress bar
224: /// It uses iTextSharp Nuget Package
225: /// </summary>
226: /// <param name="sender">The sender info (For example Main Form)</param>
227: /// <param name="e">Event Arguments</param>
228: private void btnCombineITextSharp_Click(object sender, EventArgs e)
229: {
230:     try
231:     {
232:         if (lbFiles.Items.Count < 2)
233:         {
234:             MessageBox.Show("There must be at least 2 PDF files to combine",
AppTitle, MessageBoxButtons.OK, MessageBoxIcon.Error);
235:         }
236:         else
237:         {
238:             using (var dialogExport = new FolderBrowserDialog())
239:             {
240:                 var result = dialogExport.ShowDialog();
241:                 if (result == DialogResult.OK &&
!string.IsNullOrEmpty(dialogExport.SelectedPath))
242:                 {
243:                     var outputFileName = dialogExport.SelectedPath + "/" +
Guid.NewGuid().ToString() + ".pdf";
244:                     var fileCount = lbFiles.Items.Count;
245:                     var combinedFiles = 0;
246:

```

```

247:             var outputFile = new Document();
248:             using (FileStream outputFileStream = new
FileStream(outputFileName, FileMode.Create))
249:             {
250:                 var pdfWriter = new
iTextSharp.text.pdf.PdfCopy(outputFile, outputFileStream);
251:                 if (pdfWriter == null)
252:                 {
253:                     return;
254:                 }
255:                 pdfWriter.SetFullCompression();
256:                 outputFile.Open();
257:                 for (int i = 0; i < lbFiles.Items.Count; i++)
258:                 {
259:                     try
260:                     {
261:                         var pdfReader = new
iTextSharp.text.pdf.PdfReader(lbFiles.Items[i].ToString());
262:                         pdfReader.ConsolidateNamedDestinations();
263:                         for (int j = 1; j <= pdfReader.NumberOfPages;
j++)
264:                         {
265:                             var page =
pdfWriter.GetImportedPage(pdfReader, j);
266:                             pdfWriter.AddPage(page);
267:                         }
268:                         pdfReader.Close();
269:                         combinedFiles++;
270:                     }
271:                     catch (Exception)
272:                     {
273:                         fileCount--;
274:                     }
275:
276:                     pbFiles.Value = (combinedFiles * 100 / fileCount);
277:                     if (pbFiles.Value > pbFiles.Maximum)
278:                         pbFiles.Value = pbFiles.Maximum;
279:                 }
280:                 pdfWriter.Close();
281:                 outputFile.Close();
282:             }
283:
284:             MessageBox.Show(fileCount + " PDF files successfully
combined in " + outputFileName, AppTitle, MessageBoxButtons.OK, MessageBoxIcon.Information);
285:
286:             pbFiles.Value = pbFiles.Minimum;
287:         }
288:     }
289: }
290: }
291: catch (Exception ex)
292: {
293:     MessageBox.Show("There is an error while combining PDF files in list.
Details: " + ex.GetAllMessages(), AppTitle, MessageBoxButtons.OK, MessageBoxIcon.Error);
294:     pbFiles.Value = pbFiles.Minimum;
295: }
296: }
297: }
298: }

```

Namespaces

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

1.2.4 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 = null;
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.btnCombinePdfSharp = new System.Windows.Forms.Button();
32:            this.btnAddFolder = new System.Windows.Forms.Button();
33:            this.btnAddFile = new System.Windows.Forms.Button();
34:            this.btnClearList = new System.Windows.Forms.Button();
35:            this.lbFiles = new System.Windows.Forms.ListBox();
36:            this.lblDetails = new System.Windows.Forms.Label();
37:            this.pbFiles = new System.Windows.Forms.ProgressBar();
38:            this.btnCombineITextSharp = new System.Windows.Forms.Button();
39:            this.SuspendLayout();
40:            //
41:            // btnCombinePdfSharp
42:            //
43:            this.btnCombinePdfSharp.Location = new System.Drawing.Point(487, 12);
44:            this.btnCombinePdfSharp.Name = "btnCombinePdfSharp";
45:            this.btnCombinePdfSharp.Size = new System.Drawing.Size(169, 23);
46:            this.btnCombinePdfSharp.TabIndex = 0;
47:            this.btnCombinePdfSharp.Text = "Combine Files (PdfSharp)";
48:            this.btnCombinePdfSharp.UseVisualStyleBackColor = true;
49:            this.btnCombinePdfSharp.Click += new
System.EventHandler(this.btnCombinePdfSharp_Click);
50:            //
51:            // btnAddFolder
52:            //
53:            this.btnAddFolder.Location = new System.Drawing.Point(12, 38);
54:            this.btnAddFolder.Name = "btnAddFolder";
55:            this.btnAddFolder.Size = new System.Drawing.Size(156, 23);
56:            this.btnAddFolder.TabIndex = 2;
57:            this.btnAddFolder.Text = "Add Folder";

```

```

58:         this.btnAddFolder.UseVisualStyleBackColor = true;
59:         this.btnAddFolder.Click += new
System.EventHandler(this.btnAddFolder_Click);
60:         //
61:         // btnAddFile
62:         //
63:         this.btnAddFile.Location = new System.Drawing.Point(12, 12);
64:         this.btnAddFile.Name = "btnAddFile";
65:         this.btnAddFile.Size = new System.Drawing.Size(156, 23);
66:         this.btnAddFile.TabIndex = 3;
67:         this.btnAddFile.Text = "Add Files";
68:         this.btnAddFile.UseVisualStyleBackColor = true;
69:         this.btnAddFile.Click += new System.EventHandler(this.btnAddFile_Click);
70:         //
71:         // btnClearList
72:         //
73:         this.btnClearList.Location = new System.Drawing.Point(174, 12);
74:         this.btnClearList.Name = "btnClearList";
75:         this.btnClearList.Size = new System.Drawing.Size(90, 49);
76:         this.btnClearList.TabIndex = 4;
77:         this.btnClearList.Text = "Clear File List";
78:         this.btnClearList.UseVisualStyleBackColor = true;
79:         this.btnClearList.Click += new
System.EventHandler(this.btnClearList_Click);
80:         //
81:         // lbFiles
82:         //
83:         this.lbFiles.FormattingEnabled = true;
84:         this.lbFiles.Location = new System.Drawing.Point(12, 88);
85:         this.lbFiles.Name = "lbFiles";
86:         this.lbFiles.SelectionMode =
System.Windows.Forms.SelectionMode.MultiExtended;
87:         this.lbFiles.Size = new System.Drawing.Size(644, 186);
88:         this.lbFiles.TabIndex = 5;
89:         this.lbFiles.KeyDown += new
System.Windows.Forms.KeyEventHandler(this.lbFiles_KeyDown);
90:         //
91:         // lblDetails
92:         //
93:         this.lblDetails.AutoSize = true;
94:         this.lblDetails.Location = new System.Drawing.Point(12, 65);
95:         this.lblDetails.Name = "lblDetails";
96:         this.lblDetails.Size = new System.Drawing.Size(0, 13);
97:         this.lblDetails.TabIndex = 6;
98:         //
99:         // pbFiles
100:        //
101:        this.pbFiles.Location = new System.Drawing.Point(270, 12);
102:        this.pbFiles.Name = "pbFiles";
103:        this.pbFiles.Size = new System.Drawing.Size(211, 49);
104:        this.pbFiles.TabIndex = 7;
105:        //
106:        // btnCombineITextSharp
107:        //
108:        this.btnCombineITextSharp.Location = new System.Drawing.Point(487, 38);
109:        this.btnCombineITextSharp.Name = "btnCombineITextSharp";
110:        this.btnCombineITextSharp.Size = new System.Drawing.Size(169, 23);
111:        this.btnCombineITextSharp.TabIndex = 9;
112:        this.btnCombineITextSharp.Text = "Combine Files (iTextSharp)";
113:        this.btnCombineITextSharp.UseVisualStyleBackColor = true;
114:        this.btnCombineITextSharp.Click += new
System.EventHandler(this.btnCombineITextSharp_Click);
115:        //
116:        // FrmMain
117:        //
118:        this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
119:        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
120:        this.ClientSize = new System.Drawing.Size(668, 275);
121:        this.Controls.Add(this.btnCombineITextSharp);

```

```

122:         this.Controls.Add(this.pbFiles);
123:         this.Controls.Add(this.lblDetails);
124:         this.Controls.Add(this.lbFiles);
125:         this.Controls.Add(this.btnClearList);
126:         this.Controls.Add(this.btnAddFile);
127:         this.Controls.Add(this.btnAddFolder);
128:         this.Controls.Add(this.btnCombinePdfSharp);
129:         this.Name = "FrmMain";
130:         this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
131:         this.Text = "PDF Combiner";
132:         this.FormClosed += new
System.Windows.Forms.FormClosedEventHandler(this.FrmMain_FormClosed);
133:         this.ResumeLayout(false);
134:         this.PerformLayout();
135:
136:     }
137:
138: #endregion
139:
140:     private System.Windows.Forms.Button btnCombinePdfSharp;
141:     private System.Windows.Forms.Button btnAddFolder;
142:     private System.Windows.Forms.Button btnAddFile;
143:     private System.Windows.Forms.Button btnClearList;
144:     private System.Windows.Forms.ListBox lbFiles;
145:     private System.Windows.Forms.Label lblDetails;
146:     private System.Windows.Forms.ProgressBar pbFiles;
147:     private System.Windows.Forms.Button btnCombineITextSharp;
148: }
149: }
150:

```

Namespaces

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

1.2.5 PdfCombiner.csproj

This is file PdfCombiner.csproj.

1.2.6 PdfCombiner.sln

This is file PdfCombiner.sln.

1.2.7 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.

1.2.8 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: {
13:
14:
15:     /// <summary>
16:     /// A strongly-typed resource class, for looking up localized strings, etc.
17:     /// </summary>
18:     // This class was auto-generated by the StronglyTypedResourceBuilder
19:     // class via a tool like ResGen or Visual Studio.
20:     // To add or remove a member, edit your .ResX file then rerun ResGen
21:     // with the /str option, or rebuild your VS project.
22:
23:     [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Resources.Tools.StronglyTypedResourceBuilder",
24:         "4.0.0.0")]
25:     [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
26:     [global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]
27:     internal class Resources
28:     {
29:
30:         private static global::System.Resources.ResourceManager resourceMan;
31:
32:         private static global::System.Globalization.CultureInfo resourceCulture;
33:
34:         [global::System.Diagnostics.CodeAnalysis.SuppressMessageAttribute("Microsoft.Performance",
```

```
"CA1811:AvoidUncalledPrivateCode")]
33:         internal Resources()
34:         {
35:         }
36:
37:         /// <summary>
38:         ///     Returns the cached ResourceManager instance used by this class.
39:         /// </summary>
40:
41: [global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableState.Advanced)]
42:         internal static global::System.Resources.ResourceManager ResourceManager
43:         {
44:             get
45:             {
46:                 if ((resourceMan == null))
47:                 {
48:                     global::System.Resources.ResourceManager temp = new
49: global::System.Resources.ResourceManager("PdfCombiner.Properties.Resources",
50: typeof(Resources).Assembly);
51:                     resourceMan = temp;
52:                 }
53:                 return resourceMan;
54:             }
55:         }
56:         /// <summary>
57:         ///     Overrides the current thread's CurrentUICulture property for all
58:         ///     resource lookups using this strongly typed resource class.
59:         /// </summary>
60:
61: [global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableState.Advanced)]
62:         internal static global::System.Globalization.CultureInfo Culture
63:         {
64:             get
65:             {
66:                 return resourceCulture;
67:             }
68:             set
69:             {
70:                 resourceCulture = value;
71:             }
72:         }
73:     }
```

Namespaces

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

1.2.9 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:
15:     [global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]
16:     [global::System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.VisualStudio.Editors.SettingsDesigner.SettingsSingleFileGenerator",
17: "11.0.0.0")]
18:     internal sealed partial class Settings :
19:         global::System.Configuration.ApplicationSettingsBase
20:     {
21:
22:         private static Settings defaultInstance =
23:             ((Settings)(global::System.Configuration.ApplicationSettingsBase.Synchronized(new
24: Settings())));
25:
26:         public static Settings Default
27:         {
28:             get
29:             {
30:                 return defaultInstance;
31:             }
32:         }
33:     }
34: }
```

Namespaces

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

Index

A

AssemblyInfo.cs 18

B

BaseException class 4
 about BaseException class 4
 BaseException 5

E

ExceptionExtensions class 5
 about ExceptionExtensions class 5
 GetAllMessages 6
 Exceptions.cs 19

F

Files 18
 FrmMain class 6
 about FrmMain class 6
 AppTitle 8
 btnAddFile 8
 btnAddFile_Click 9
 btnAddFolder 8
 btnAddFolder_Click 10
 btnClearList 8
 btnClearList_Click 10
 btnCombineITextSharp 8
 btnCombineITextSharp_Click 11
 btnCombinePdfSharp 8
 btnCombinePdfSharp_Click 12
 components 8
 Dispose 13
 FrmMain 7
 FrmMain_FormClosed 14
 InitializeComponent 14
 InitializeFileDialog 16
 lbFiles 9
 lbFiles_KeyDown 16

lbIDetails 9
 pbFiles 9
 FrmMain.cs 20
 FrmMain.Designer.cs 25

P

PdfCombiner 1
 PdfCombiner namespace 1
 Classes 4
 PdfCombiner.BaseException 4
 PdfCombiner.BaseException.BaseException 5
 PdfCombiner.csproj 27
 PdfCombiner.ExceptionExtensions 5
 PdfCombiner.ExceptionExtensions.GetAllMessages 6
 PdfCombiner.FrmMain 6
 PdfCombiner.FrmMain.AppTitle 8
 PdfCombiner.FrmMain.btnAddFile 8
 PdfCombiner.FrmMain.btnAddFile_Click 9
 PdfCombiner.FrmMain.btnAddFolder 8
 PdfCombiner.FrmMain.btnAddFolder_Click 10
 PdfCombiner.FrmMain.btnClearList 8
 PdfCombiner.FrmMain.btnClearList_Click 10
 PdfCombiner.FrmMain.btnCombineITextSharp 8
 PdfCombiner.FrmMain.btnCombineITextSharp_Click 11
 PdfCombiner.FrmMain.btnCombinePdfSharp 8
 PdfCombiner.FrmMain.btnCombinePdfSharp_Click 12
 PdfCombiner.FrmMain.components 8
 PdfCombiner.FrmMain.Dispose 13
 PdfCombiner.FrmMain.FrmMain 7
 PdfCombiner.FrmMain.FrmMain_FormClosed 14
 PdfCombiner.FrmMain.InitializeComponent 14
 PdfCombiner.FrmMain.InitializeFileDialog 16
 PdfCombiner.FrmMain.lbFiles 9
 PdfCombiner.FrmMain.lbFiles_KeyDown 16
 PdfCombiner.FrmMain.lbIDetails 9
 PdfCombiner.FrmMain.pbFiles 9
 PdfCombiner.Program 17
 PdfCombiner.Program.Main 17
 PdfCombiner.Properties 1
 PdfCombiner.Properties namespace 1

Classes 1

PdfCombiner.Properties.Resources 1
PdfCombiner.Properties.Resources.Culture 3
PdfCombiner.Properties.Resources.resourceCulture 2
PdfCombiner.Properties.Resources.resourceMan 3
PdfCombiner.Properties.Resources.ResourceManager 3
PdfCombiner.Properties.Resources.Resources 2
PdfCombiner.Properties.Settings 3
PdfCombiner.Properties.Settings.Default 4
PdfCombiner.Properties.Settings.defaultInstance 4
PdfCombiner.sln 27
Program class 17
 about Program class 17
 Main 17
Program.cs 27

R

Resources class 1
 about Resources class 1
 Culture 3
 resourceCulture 2
 resourceMan 3
 ResourceManager 3
 Resources 2
Resources.Designer.cs 28

S

Settings class 3
 about Settings class 3
 Default 4
 defaultInstance 4
Settings.Designer.cs 29