made with doc-o-matic

PDF Combiner

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

Copyright (c) 2022. All rights reserved.

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

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

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.

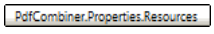
Classes

	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



Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

C#

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Resources.Tools.StronglyTyped
dResourceBuilder",
"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
S	resourceCulture (see page 2)	This is resourceCulture, a member of class Resources.
S	resourceMan (see page 3)	This is resourceMan, a member of class Resources.

Resources Properties

	Name	Description
S	AppTitle (see page 3)	Looks up a localized string similar to PDF Birlestirici.
S	Culture (see page 3)	Overrides the current thread's CurrentUICulture property for all resource lookups using this strongly typed resource class.
S	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;
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

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.Editor
BrowsableState.Advanced)]
internal static global::System.Globalization.CultureInfo Culture;
```

1.1.1.1.1.3.3 Resources.ResourceManager Property

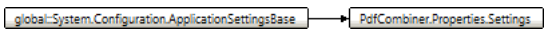
Returns the cached ResourceManager instance used by this class.

C#

```
[global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.Editor
BrowsableState.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.Sett
ingsDesigner.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.

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

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.

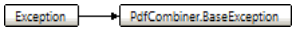
Classes

	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

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

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

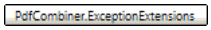
This is BaseException, a member of class BaseException.

Body Source

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

1.1.2.2 ExceptionExtensions Class

Class Hierarchy



C#

```
public static class ExceptionExtensions;
```

File

ExceptionExtensions.cs (see page 29)

Description

This is class PdfCombiner.ExceptionExtensions.

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

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.

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

FrmMain Fields

	Name	Description
S R	_languageList (see page 8)	Culture lists whose resources have been added to project
S	_resource (see page 9)	This is _resource, a member of class FrmMain.
	btnAddFile (see page 9)	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 10)	This is cmbLanguage, a member of class FrmMain.
	components (see page 10)	Required designer variable.
	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 11)	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.

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 13)	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 9) file And with this, you can see form elements, info messages etc. with the selected _resource (see page 9) 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.

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

	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.
S	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
S	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
S	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._languageList Field

C#

```
private static readonly List<string> _languageList = new List<string> {"EN", "TR", "DE",
"FR", "RU", "ES", "IT", "ZH", "AR", "NL", "PT", "IN", "ID", "JP", "BG"};
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

Description

Culture lists whose resources have been added to project

1.1.2.3.2.2 FrmMain._resource Field

C#

```
private static ResourceManager _resource;
```

Description

This is _resource, a member of class FrmMain.

1.1.2.3.2.3 FrmMain.btnAddFile Field

C#

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

Description

This is btnAddFile, a member of class FrmMain.

1.1.2.3.2.4 FrmMain.btnAddFolder Field

C#

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

Description

This is btnAddFolder, a member of class FrmMain.

1.1.2.3.2.5 FrmMain.btnClearList Field

C#

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

Description

This is btnClearList, a member of class FrmMain.

1.1.2.3.2.6 FrmMain.btnCombineITextSharp Field

C#

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

Description

This is btnCombineITextSharp, a member of class FrmMain.

1.1.2.3.2.7 FrmMain.btnCombinePdfSharp Field

C#

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

Description

This is btnCombinePdfSharp, a member of class FrmMain.

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

1.1.2.3.2.8 FrmMain.cmbLanguage Field

C#

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

Description

This is cmbLanguage, a member of class FrmMain.

1.1.2.3.2.9 FrmMain.components Field

Required designer variable.

C#

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

1.1.2.3.2.10 FrmMain.lbFiles Field

C#

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

Description

This is lbFiles, a member of class FrmMain.

1.1.2.3.2.11 FrmMain.lblDetails Field

C#

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

Description

This is lblDetails, a member of class FrmMain.

1.1.2.3.2.12 FrmMain.menuItemDelete Field

C#

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

Description

This is menuItemDelete, a member of class FrmMain.

1.1.2.3.2.13 FrmMain.menuItemOrderByNameAscending Field

C#

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

Description

This is menuItemOrderByNameAscending, a member of class FrmMain.

1.1.2.3.2.14 FrmMain.menuItemOrderByNameDescending Field

C#

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

Description

This is menuItemOrderByNameDescending, a member of class FrmMain.

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

1.1.2.3.2.15 FrmMain.menuItemOrderByPathAscending Field

C#

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

Description

This is menuItemOrderByPathAscending, a member of class FrmMain.

1.1.2.3.2.16 FrmMain.menuItemOrderByPathDescending Field

C#

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

Description

This is menuItemOrderByPathDescending, a member of class FrmMain.

1.1.2.3.2.17 FrmMain.menuStrip Field

C#

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

Description

This is menuStrip, a member of class FrmMain.

1.1.2.3.2.18 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();
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

```
 7:          var addedFileCount = 0;
 8:          if (result != DialogResult.OK || dialogAddFile.FileNames == null) return;
 9:          foreach (var file in dialogAddFile.FileNames)
10:          {
11:              if (lbFiles.Items.Contains(file)) continue;
12:              lbFiles.Items.Add(file);
13:              addedFileCount++;
14:
15:              pbFiles.Value = addedFileCount * 100 / dialogAddFile.FileNames.Length;
16:              if (pbFiles.Value > pbFiles.Maximum)
17:                  pbFiles.Value = pbFiles.Maximum;
18:              ActiveForm.Text = "%" + pbFiles.Value;
19:          }
20:
21:          MessageBox.Show(addedFileCount + _resource.GetString("FileAddMessage"),
22:              _resource.GetString("AppTitle"), MessageBoxButtons.OK,
MessageBoxIcon.Information);
23:
24:          pbFiles.Value = pbFiles.Minimum;
25:          ActiveForm.Text = _resource.GetString("AppTitle");
26:      }
27: }
```

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)) return;
 7:         var fileNames = Directory.GetFiles(dialogAddFolder.SelectedPath, "*.pdf",
 8:             SearchOption.AllDirectories);
 9:         var addedFileCount = 0;
10:         foreach (var file in fileNames)
11:         {
12:             if (lbFiles.Items.Contains(file)) continue;
13:             lbFiles.Items.Add(file);
14:             addedFileCount++;
15:
16:             pbFiles.Value = addedFileCount * 100 / fileNames.Length;
17:             if (pbFiles.Value > pbFiles.Maximum)
18:                 pbFiles.Value = pbFiles.Maximum;
19:             ActiveForm.Text = "%" + pbFiles.Value;
20:         }
21:
22:         MessageBox.Show(addedFileCount + _resource.GetString("FileAddMessage"),
23:             _resource.GetString("AppTitle"), MessageBoxButtons.OK,
MessageBoxIcon.Information);
24:
25:         pbFiles.Value = pbFiles.Minimum;
26:         ActiveForm.Text = _resource.GetString("AppTitle");
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

```
27:     }
28: }
```

1.1.2.3.3.3 FrmMain.btnClearList_Click Method

This function is used to clear the file list in listbox

C#

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

Parameters

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

Body Source

```
1: private void btnClearList_Click(object sender, EventArgs e)
2: {
3:     var fileCount = lbFiles.Items.Count;
4:     lbFiles.Items.Clear();
5:     MessageBox.Show(fileCount + _resource.GetString("DeleteFileMessage"),
_resource.GetString("AppTitle"),
6:         MessageBoxButtons.OK, MessageBoxIcon.Information);
7: }
```

1.1.2.3.3.4 FrmMain.btnCombineITextSharp_Click Method

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

C#

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

Parameters

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

Body Source

```
 1: private void btnCombineITextSharp_Click(object sender, EventArgs e)
 2: {
 3:     try
 4:     {
 5:         if (lbFiles.Items.Count < 2)
 6:             MessageBox.Show(_resource.GetString("CombineWarning"),
_resource.GetString("AppTitle"),
 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)) return;
14:                 var outputFileName = dialogExport.SelectedPath + "/" + Guid.NewGuid() +
".pdf";
15:                 var fileCount = lbFiles.Items.Count;
16:                 var combinedFiles = 0;
17:
18:                 var outputFile = new Document();
19:                 using (var outputFileStream = new FileStream(outputFileName,
FileMode.Create))
20:                 {
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

```
21:                     using (var pdfWriter = new iTextSharp.text.pdf.PdfCopy(outputFile,
outputFileStream))
22:                     {
23:                         if (pdfWriter == null)
24:                         {
25:                             return;
26:                         }
27:
28:                         pdfWriter.SetFullCompression();
29:                         outputFile.Open();
30:                         foreach (var t in lbFiles.Items)
31:                         {
32:                             try
33:                             {
34:                                 var pdfReader =
35:                                     new iTextSharp.text.pdf.PdfReader(t.ToString());
36:                                 pdfReader.ConsolidateNamedDestinations();
37:                                 for (var j = 1; j <= pdfReader.NumberOfPages; j++)
38:                                 {
39:                                     var page = pdfWriter.GetImportedPage(pdfReader, j);
40:                                     pdfWriter.AddPage(page);
41:                                 }
42:
43:                                 pdfReader.Close();
44:                                 combinedFiles++;
45:                             }
46:                             catch (Exception)
47:                             {
48:                                 fileCount--;
49:                             }
50:
51:                             pbFiles.Value = combinedFiles * 100 / fileCount;
52:                             if (pbFiles.Value > pbFiles.Maximum)
53:                                 pbFiles.Value = pbFiles.Maximum;
54:                             ActiveForm.Text = "%" + pbFiles.Value;
55:                         }
56:
57:                         pdfWriter.Close();
58:                     }
59:
60:                     outputFile.Close();
61:                 }
62:
63:                 MessageBox.Show(fileCount + _resource.GetString("CombineFileMessage") +
outputFileName,
64:                     _resource.GetString("AppTitle"), MessageBoxButtons.OK,
MessageBoxIcon.Information);
65:
66:                 pbFiles.Value = pbFiles.Minimum;
67:                 ActiveForm.Text = _resource.GetString("AppTitle");
68:             }
69:         }
70:     }
71:     catch (Exception ex)
72:     {
73:         MessageBox.Show(_resource.GetString("CombineErrorMessage") +
ex.GetAllMessages(),
74:             _resource.GetString("AppTitle"), MessageBoxButtons.OK,
MessageBoxIcon.Error);
75:         pbFiles.Value = pbFiles.Minimum;
76:         ActiveForm.Text = _resource.GetString("AppTitle");
77:     }
78: }
```

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

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

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)) return;
14:                 var outputFileName = dialogExport.SelectedPath + "/" + Guid.NewGuid() +
".pdf";
15:                 using (var outputFile = new PdfDocument())
16:                 {
17:                     outputFile.Options.CompressContentStreams = true;
18:                     outputFile.Options.EnableCcittCompressionForBilevelImages = true;
19:                     outputFile.Options.FlateEncodeMode =
PdfFlateEncodeMode.BestCompression;
20:                     var fileCount = lbFiles.Items.Count;
21:                     var combinedFiles = 0;
22:
23:                     foreach (var t in lbFiles.Items)
24:                     {
25:                         try
26:                         {
27:                             var inputDocument = PdfReader.Open(t.ToString(),
28:                                 PdfDocumentOpenMode.Import);
29:                             var count = inputDocument.PageCount;
30:                             for (var idx = 0; idx < count; idx++)
31:                             {
32:                                 var page = inputDocument.Pages[idx];
33:                                 outputFile.AddPage(page);
34:                             }
35:
36:                             inputDocument.Close();
37:                             combinedFiles++;
38:                         }
39:                         catch (Exception)
40:                         {
41:                             fileCount--;
42:                         }
43:
44:                         pbFiles.Value = combinedFiles * 100 / fileCount;
45:                         if (pbFiles.Value > pbFiles.Maximum)
46:                             pbFiles.Value = pbFiles.Maximum;
47:                         ActiveForm.Text = "%" + pbFiles.Value;
48:                     }
49:
50:                     outputFile.Save(outputFileName);
51:                     outputFile.Close();
52:
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

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

1.1.2.3.3.6 FrmMain.cmbLanguage_SelectedIndexChanged Method

This function is used to set the _resource (see page 9) file And with this, you can see form elements, info messages etc. with the selected _resource (see page 9) 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) return;
 4:     try
 5:     {
 6:         _resource = new ResourceManager(
 7:             "PdfCombiner.Resources.AppResources-" +
 8:             cmbLanguage.SelectedItem.ToString().ToLower(new CultureInfo("en-US")),
 9:             Assembly.GetExecutingAssembly());
10:         FormMembersNameInitialization();
11:     }
12:     catch (Exception)
13:     {
14:         _resource = new ResourceManager("PdfCombiner.Resources.AppResources-en",
15:             Assembly.GetExecutingAssembly());
16:         FormMembersNameInitialization();
17:     }
18: }
```

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();
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

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:     foreach (var t in lbFiles.SelectedItems)
 7:     {
 8:         removedItems.Add(t.ToString());
 9:     }
10:
11:     foreach (var item in lbFiles.Items)
12:     {
13:         if (removedItems.All(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");
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

```
 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"), MessageBoxButtons.OK,
4:         MessageBoxIcon.Information);
5:     Application.Exit();
6: }
```

1.1.2.3.3.11 FrmMain.FrmMain_Shown Method

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

C#

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

Parameters

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

Body Source

```
 1: private void FrmMain_Shown(object sender, EventArgs e)
 2: {
 3:     lbFiles.ContextMenuStrip = menuStrip;
 4:     cmbLanguage.DropDownStyle = ComboBoxStyle.DropDownList;
 5:     cmbLanguage.Items.Clear();
 6:     foreach (var item in _languageList)
 7:         cmbLanguage.Items.Add(item);
 8:     _resource = new ResourceManager("PdfCombiner.Resources.AppResources-tr",
Assembly.GetExecutingAssembly());
 9:     FormMembersNameInitialization();
10: }
```

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();
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

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:     //
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

```
 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";
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

```
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.Font = new System.Drawing.Font("Microsoft Sans Serif", 15F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte) (162)));
167:     this.cmbLanguage.FormattingEnabled = true;
168:     this.cmbLanguage.Items.AddRange(new object[] {"TR", "EN", "DE", "FR", "RU", "ES"});
169:     this.cmbLanguage.Location = new System.Drawing.Point(374, 6);
170:     this.cmbLanguage.Name = "cmbLanguage";
171:     this.cmbLanguage.Size = new System.Drawing.Size(62, 33);
172:     this.cmbLanguage.TabIndex = 10;
173:     this.cmbLanguage.Text = "Select Language";
174:     this.cmbLanguage.SelectedIndexChanged += new
System.EventHandler(this.cmbLanguage_SelectedIndexChanged);
175:     //
176:     // FrmMain
177:     //
178:     this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

```
179:     this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
180:     this.ClientSize = new System.Drawing.Size(668, 287);
181:     this.Controls.Add(this.cmbLanguage);
182:     this.Controls.Add(this.btnCombineITextSharp);
183:     this.Controls.Add(this.pbFiles);
184:     this.Controls.Add(this.lblDetails);
185:     this.Controls.Add(this.lbFiles);
186:     this.Controls.Add(this.btnClearList);
187:     this.Controls.Add(this.btnAddFile);
188:     this.Controls.Add(this.btnAddFolder);
189:     this.Controls.Add(this.btnCombinePdfSharp);
190:     this.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte) (162)));
191:     this.Name = "FrmMain";
192:     this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
193:     this.Text = "PDF Combiner";
194:     this.FormClosed += new
System.Windows.Forms.FormClosedEventHandler(this.FrmMain_FormClosed);
195:     this.Shown += new System.EventHandler(this.FrmMain_Shown);
196:     this.menuStrip.ResumeLayout(false);
197:     this.ResumeLayout(false);
198:     this.PerformLayout();
199: }
```

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: {
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

```
 3:      var point = lbFiles.PointToClient(new Point(e.X, e.Y));
 4:      var index = lbFiles.IndexFromPoint(point);
 5:      if (index < 0) index = 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:      lbFiles.Items.Remove(data);
 9:      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) return;
 8:     for (var i = 0; i < lbFiles.Items.Count; i++)
 9:     {
10:         lbFiles.SetSelected(i, true);
11:     }
12: }
```

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);
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

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)) return;
4:     if (lbFiles.SelectedItem == null) return;
5:     lbFiles.DoDragDrop(lbFiles.SelectedItem, DragDropEffects.Move);
6: }
```

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"),
_resource.GetString("AppTitle"),
12:                     MessageBoxButtons.YesNo, MessageBoxIcon.Warning);
13:             if (deleteDialog == DialogResult.Yes)
14:                 DeleteFilesFromListBox();
15:         }
16:         else
17:             MessageBox.Show(_resource.GetString("NoSelectedFile"),
_resource.GetString("AppTitle"),
18:                 MessageBoxButtons.OK, MessageBoxIcon.Error);
19:     }
20:     else
21:         MessageBox.Show(_resource.GetString("NoFile"), _resource.GetString("AppTitle"),
MessageBoxButtons.OK,
22:             MessageBoxIcon.Error);
23: }
```

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);
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

Parameters

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

Body Source

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

1.1.2.3.3.20 FrmMain.menuItemOrderByNameDescending_Click Method

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

C#

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

Parameters

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

Body Source

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

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

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

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

1.1.2.3.3.23 FrmMain.SortItemsByName Method

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

C#

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

Parameters

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

Body Source

```
 1: private static ListBox SortItemsByName(ListBox listBox, bool ascending)
 2: {
 3:     var fileNameList = new List<string>();
 4:     var items = listBox.Items.OfType<object>().ToList();
 5:     const char c = '\u005c';
 6:     foreach (var item in items)
 7:     {
 8:         var fullPath = item.ToString();
 9:         var list = fullPath.Split(c).ToList();
10:         if (list.Count > 0)
11:             fileNameList.Add(list[list.Count - 1]);
12:     }
13:
14:     fileNameList = ascending ? fileNameList.OrderBy(j => j).ToList() :
fileNameList.OrderByDescending(j => j).ToList();
15:
16:     listBox.Items.Clear();
17:     foreach (var item in fileNameList)
18:         listBox.Items.Add(items.First(j => j.ToString().EndsWith(item)));
19:
20:     return listBox;
21: }
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

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

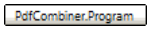
Parameters	Description
ListBox listBox	ListBox Info
bool ascending	Order Type is Ascending Or Not

Body Source

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

1.1.2.4 Program Class

Class Hierarchy



C#

```
internal 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: private static void Main()
3: {
4:     Application.EnableVisualStyles();
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

```
 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.InteropServices;
 3:
 4: // General Information about an assembly is controlled through the following
 5: // set of attributes. Change these attribute values to modify the information
 6: // associated with an assembly.
 7: [assembly: AssemblyTitle("PdfCombiner")]
 8: [assembly: AssemblyDescription("")]
 9: [assembly: AssemblyConfiguration("")]
10: [assembly: AssemblyCompany("")]
11: [assembly: AssemblyProduct("PdfCombiner")]
12: [assembly: AssemblyCopyright("Copyright ©  2022")]
13: [assembly: AssemblyTrademark("")]
14: [assembly: AssemblyCulture("")]
15:
16: // Setting ComVisible to false makes the types in this assembly not visible
17: // to COM components.  If you need to access a type in this assembly from
18: // COM, set the ComVisible attribute to true on that type.
19: [assembly: ComVisible(false)]
20:
21: // The following GUID is for the ID of the typelib if this project is exposed to COM
22: [assembly: Guid("48e34ee8-dfaa-48db-acfd-70573cee9c00")]
23:
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

```
24: // Version information for an assembly consists of the following four values:
25: //
26: //      Major Version
27: //      Minor Version
28: //      Build Number
29: //      Revision
30: //
31: // You can specify all the values or you can default the Build and Revision Numbers
32: // by using the '*' as shown below:
33: // [assembly: AssemblyVersion("1.0.*")]
34: [assembly: AssemblyVersion("1.0.0.0")]
35: [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) :
base(message, innerException)
21:         {
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:     {
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

```
 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:         private 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", "NL", "PT", "IN",
"ID", "JP", "BG"};
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>
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

```
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) return;
70:             try
71:             {
72:                 _resource = new ResourceManager(
73:                     "PdfCombiner.Resources.AppResources-" +
74:                     cmbLanguage.SelectedItem.ToString().ToLower(new
CultureInfo("en-US")),
75:                     Assembly.GetExecutingAssembly());
76:                 FormMembersNameInitialization();
77:             }
78:             catch (Exception)
79:             {
80:                 _resource = new
ResourceManager("PdfCombiner.Resources.AppResources-en",
81:                     Assembly.GetExecutingAssembly());
82:                 FormMembersNameInitialization();
83:             }
84:         }
85:
86:         /// <summary>
87:         /// This method is used to change text of form members by the
88:         /// Resource file which is selected
89:         /// </summary>
90:         private void FormMembersNameInitialization()
91:         {
92:             ActiveForm.Text = _resource.GetString("AppTitle");
93:             btnAddFile.Text = _resource.GetString("AddFile");
94:             btnAddFolder.Text = _resource.GetString("AddFolder");
95:             btnClearList.Text = _resource.GetString("ClearFileList");
96:             btnCombineITextSharp.Text = _resource.GetString("CombineFilesITextSharp");
97:             btnCombinePdfSharp.Text = _resource.GetString("CombineFilesPdfSharp");
98:             menuItemDelete.Text = _resource.GetString("Delete");
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

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

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

```
161:         /// This function is used to add PDF files in a folder which you choose in
162:         /// Folder Dialog recursively
163:         /// When you use it the added files will be seen in listbox
164:         /// </summary>
165:         /// <param name="sender">The sender info (For example Main Form)</param>
166:         /// <param name="e">Event Arguments</param>
167:         private void btnAddFolder_Click(object sender, EventArgs e)
168:         {
169:             using (var dialogAddFolder = new FolderBrowserDialog())
170:             {
171:                 var result = dialogAddFolder.ShowDialog();
172:                 if (result != DialogResult.OK ||
string.IsNullOrEmpty(dialogAddFolder.SelectedPath)) return;
173:                 var fileNames = Directory.GetFiles(dialogAddFolder.SelectedPath,
"*.pdf",
174:                     SearchOption.AllDirectories);
175:                 var addedFileCount = 0;
176:                 foreach (var file in fileNames)
177:                 {
178:                     if (lbFiles.Items.Contains(file)) continue;
179:                     lbFiles.Items.Add(file);
180:                     addedFileCount++;
181:
182:                     pbFiles.Value = addedFileCount * 100 / fileNames.Length;
183:                     if (pbFiles.Value > pbFiles.Maximum)
184:                         pbFiles.Value = pbFiles.Maximum;
185:                     ActiveForm.Text = "%" + pbFiles.Value;
186:                 }
187:
188:                 MessageBox.Show(addedFileCount + _resource.GetString("FileAddMessage"),
189:                     _resource.GetString("AppTitle"), MessageBoxButtons.OK,
MessageBoxIcon.Information);
190:
191:                 pbFiles.Value = pbFiles.Minimum;
192:                 ActiveForm.Text = _resource.GetString("AppTitle");
193:             }
194:         }
195:
196:         #endregion
197:
198:         #region Combine PDF Methods
199:
200:         /// <summary>
201:         /// This function is used to take list of PDF files in listbox,
202:         /// Combine them with a single file in location which you choose in folder
dialog
203:         /// And you can see the progress in progress bar
204:         /// It uses PdfSharp Nuget Package
205:         /// </summary>
206:         /// <param name="sender">The sender info (For example Main Form)</param>
207:         /// <param name="e">Event Arguments</param>
208:         private void btnCombinePdfSharp_Click(object sender, EventArgs e)
209:         {
210:             try
211:             {
212:                 if (lbFiles.Items.Count < 2)
213:                     MessageBox.Show(_resource.GetString("CombineWarning"),
_resource.GetString("AppTitle"),
214:                         MessageBoxButtons.OK, MessageBoxIcon.Error);
215:                 else
216:                 {
217:                     using (var dialogExport = new FolderBrowserDialog())
218:                     {
219:                         var result = dialogExport.ShowDialog();
220:                         if (result != DialogResult.OK ||
string.IsNullOrEmpty(dialogExport.SelectedPath)) return;
221:                         var outputFileName = dialogExport.SelectedPath + "/" +
Guid.NewGuid() + ".pdf";
222:                         using (var outputFile = new PdfDocument())
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

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

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

```
285:         /// <param name="e">Event Arguments</param>
286:         private void btnCombineITextSharp_Click(object sender, EventArgs e)
287:         {
288:             try
289:             {
290:                 if (lbFiles.Items.Count < 2)
291:                     MessageBox.Show(_resource.GetString("CombineWarning"),
_resource.GetString("AppTitle"),
292:                         MessageBoxButtons.OK, MessageBoxIcon.Error);
293:                 else
294:                 {
295:                     using (var dialogExport = new FolderBrowserDialog())
296:                     {
297:                         var result = dialogExport.ShowDialog();
298:                         if (result != DialogResult.OK ||
string.IsNullOrEmpty(dialogExport.SelectedPath)) return;
299:                         var outputFileName = dialogExport.SelectedPath + "/" +
Guid.NewGuid() + ".pdf";
300:                         var fileCount = lbFiles.Items.Count;
301:                         var combinedFiles = 0;
302:
303:                         var outputFile = new Document();
304:                         using (var outputFileStream = new FileStream(outputFileName,
FileMode.Create))
305:                         {
306:                             using (var pdfWriter = new
iTextSharp.text.pdf.PdfCopy(outputFile, outputFileStream))
307:                             {
308:                                 if (pdfWriter == null)
309:                                 {
310:                                     return;
311:                                 }
312:
313:                                 pdfWriter.SetFullCompression();
314:                                 outputFile.Open();
315:                                 foreach (var t in lbFiles.Items)
316:                                 {
317:                                     try
318:                                     {
319:                                         var pdfReader =
320:                                             new
iTextSharp.text.pdf.PdfReader(t.ToString());
321:                                         pdfReader.ConsolidateNamedDestinations();
322:                                         for (var j = 1; j <= pdfReader.NumberOfPages;
j++)
323:                                         {
324:                                             var page =
pdfWriter.GetImportedPage(pdfReader, j);
325:                                             pdfWriter.AddPage(page);
326:                                         }
327:
328:                                         pdfReader.Close();
329:                                         combinedFiles++;
330:                                     }
331:                                     catch (Exception)
332:                                     {
333:                                         fileCount--;
334:                                     }
335:
336:                                     pbFiles.Value = combinedFiles * 100 / fileCount;
337:                                     if (pbFiles.Value > pbFiles.Maximum)
338:                                         pbFiles.Value = pbFiles.Maximum;
339:                                     ActiveForm.Text = "%" + pbFiles.Value;
340:                                 }
341:
342:                                 pdfWriter.Close();
343:                             }
344:
345:                             outputFile.Close();
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

```
346:                         }
347:
348:                         MessageBox.Show(fileCount +
_resource.GetString("CombineFileMessage") + outputFileName,
349:                             _resource.GetString("AppTitle"), MessageBoxButtons.OK,
MessageBoxIcon.Information);
350:
351:                         pbFiles.Value = pbFiles.Minimum;
352:                         ActiveForm.Text = _resource.GetString("AppTitle");
353:                     }
354:                 }
355:             }
356:             catch (Exception ex)
357:             {
358:                 MessageBox.Show(_resource.GetString("CombineErrorMessage") +
ex.GetAllMessages(),
359:                     _resource.GetString("AppTitle"), MessageBoxButtons.OK,
MessageBoxIcon.Error);
360:                 pbFiles.Value = pbFiles.Minimum;
361:                 ActiveForm.Text = _resource.GetString("AppTitle");
362:             }
363:         }
364:
365:         #endregion
366:
367:         #region Delete Item Methods
368:
369:         /// <summary>
370:         /// This function is used to operate some process in files in listbox which
are chosen
371:         /// </summary>
372:         /// <param name="sender">The sender info (For example Main Form)</param>
373:         /// <param name="e">Event Arguments</param>
374:         private void lbFiles_KeyDown(object sender, KeyEventArgs e)
375:         {
376:             // this is used to delete selected files in listbox when you press Delete
key
377:             if (e.KeyValue == Keys.Delete.GetHashCode())
378:                 DeleteFilesFromListBox();
379:             // This is used to select all items in listbox when you press Ctrl + A key
combination
380:             if (!e.Control || e.KeyCode != Keys.A) return;
381:             for (var i = 0; i < lbFiles.Items.Count; i++)
382:             {
383:                 lbFiles.SetSelected(i, true);
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)
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

```
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:                foreach (var t in lbFiles.SelectedItems)
428:                {
429:                    removedItems.Add(t.ToString());
430:                }
431:
432:                foreach (var item in lbFiles.Items)
433:                {
434:                    if (removedItems.All(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)
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

```
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:             var items = listBox.Items.OfType<object>().ToList();
530:             listBox.Items.Clear();
531:             listBox.Items.AddRange(ascending
532:                 ? items.OrderBy(i => i).ToArray()
533:                 : items.OrderByDescending(i => i).ToArray());
534:             return listBox;
535:         }
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

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

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

```
604:             var data = e.Data.GetData(typeof(string));
605:             lbFiles.Items.Remove(data);
606:             lbFiles.Items.Insert(index, data);
607:         }
608:
609:         #endregion
610:     }
611: }
```

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();
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

```
 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;
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

```
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";
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

```
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.Font = new System.Drawing.Font("Microsoft Sans Serif",
15F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte) (162)));
195:             this.cmbLanguage.FormattingEnabled = true;
196:             this.cmbLanguage.Items.AddRange(new object[] {"TR", "EN", "DE", "FR",
"RU", "ES"});
197:             this.cmbLanguage.Location = new System.Drawing.Point(374, 6);
198:             this.cmbLanguage.Name = "cmbLanguage";
199:             this.cmbLanguage.Size = new System.Drawing.Size(62, 33);
200:             this.cmbLanguage.TabIndex = 10;
201:             this.cmbLanguage.Text = "Select Language";
202:             this.cmbLanguage.SelectedIndexChanged += new
System.EventHandler(this.cmbLanguage_SelectedIndexChanged);
203:             //
204:             // FrmMain
205:             //
206:             this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
207:             this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
208:             this.ClientSize = new System.Drawing.Size(668, 287);
209:             this.Controls.Add(this.cmbLanguage);
210:             this.Controls.Add(this.btnCombineITextSharp);
211:             this.Controls.Add(this.pbFiles);
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

```
212:             this.Controls.Add(this.lblDetails);
213:             this.Controls.Add(this.lbFiles);
214:             this.Controls.Add(this.btnClearList);
215:             this.Controls.Add(this.btnAddFile);
216:             this.Controls.Add(this.btnAddFolder);
217:             this.Controls.Add(this.btnCombinePdfSharp);
218:             this.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte) (162)));
219:             this.Name = "FrmMain";
220:             this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
221:             this.Text = "PDF Combiner";
222:             this.FormClosed += new
System.Windows.Forms.FormClosedEventHandler(this.FrmMain_FormClosed);
223:             this.Shown += new System.EventHandler(this.FrmMain_Shown);
224:             this.menuStrip.ResumeLayout(false);
225:             this.ResumeLayout(false);
226:             this.PerformLayout();
227:         }
228:
229:         public System.Windows.Forms.ComboBox cmbLanguage;
230:
231:         #endregion
232:
233:         private System.Windows.Forms.Button btnCombinePdfSharp;
234:         private System.Windows.Forms.Button btnAddFolder;
235:         private System.Windows.Forms.Button btnAddFile;
236:         private System.Windows.Forms.Button btnClearList;
237:         private System.Windows.Forms.ListBox lbFiles;
238:         private System.Windows.Forms.Label lblDetails;
239:         private System.Windows.Forms.ProgressBar pbFiles;
240:         private System.Windows.Forms.Button btnCombineITextSharp;
241:         private System.Windows.Forms.ContextMenuStrip menuStrip;
242:         private System.Windows.Forms.ToolStripMenuItem menuItemDelete;
243:         private System.Windows.Forms.ToolStripMenuItem menuItemOrderByPathAscending;
244:         private System.Windows.Forms.ToolStripMenuItem menuItemOrderByPathDescending;
245:         private System.Windows.Forms.ToolStripMenuItem menuItemOrderByNameAscending;
246:         private System.Windows.Forms.ToolStripMenuItem menuItemOrderByNameDescending;
247:     }
248: }
249:
```

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.

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

1.2.8 Program.cs

This is file Program.cs.

Body Source

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

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:
```

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Resources.Tools.StronglyTyped
ResourceBuilder",
"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.Editor
BrowsableState.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.Editor
BrowsableState.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.

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

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.Sett
ingsDesigner.SettingsSingleFileGenerator",
"11.0.0.0")]
16:     internal sealed partial class Settings :
global::System.Configuration.ApplicationSettingsBase
17:     {
18:
19:         private static Settings defaultInstance =
((Settings)(global::System.Configuration.ApplicationSettingsBase.Synchronized(new
Settings())));
20:     }
21: }
```

Namespaces

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

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

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
- _languageList 8
- _resource 9
- about FrmMain class 6
- btnAddFile 9
- btnAddFile_Click 11
- btnAddFolder 9
- btnAddFolder_Click 12
- btnClearList 9
- btnClearList_Click 13
- btnCombineITextSharp 9
- btnCombineITextSharp_Click 13
- btnCombinePdfSharp 9
- btnCombinePdfSharp_Click 14
- cmbLanguage 10
- cmbLanguage_SelectedIndexChanged 16
- components 10
- DeleteFilesFromListBox 16
- Dispose 17
- FormMembersNameInitialization 17
- FrmMain 8
- FrmMain_FormClosed 18
- FrmMain_Shown 18
- InitializeComponent 18
- InitializeFileDialog 22
- lbFiles 10
- lbFiles_DragDrop 22
- lbFiles_DragOver 23
- lbFiles_KeyDown 23
- lbFiles_MouseDown 23
- lblDetails 10
- menuItemDelete 10
- menuItemDelete_Click 24
- menuItemOrderByNameAscending 10
- menuItemOrderByNameAscending_Click 24
- menuItemOrderByNameDescending 10
- menuItemOrderByNameDescending_Click 25
- menuItemOrderByPathAscending 11
- menuItemOrderByPathAscending_Click 25
- menuItemOrderByPathDescending 11
- menuItemOrderByPathDescending_Click 26
- menuStrip 11
- pbFiles 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._languageList 8

PdfCombiner.FrmMain._resource 9

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

PdfCombiner.FrmMain.btnAddFile 9
PdfCombiner.FrmMain.btnAddFile_Click 11
PdfCombiner.FrmMain.btnAddFolder 9
PdfCombiner.FrmMain.btnAddFolder_Click 12
PdfCombiner.FrmMain.btnClearList 9
PdfCombiner.FrmMain.btnClearList_Click 13
PdfCombiner.FrmMain.btnCombineITextSharp 9
PdfCombiner.FrmMain.btnCombineITextSharp_Click 13
PdfCombiner.FrmMain.btnCombinePdfSharp 9
PdfCombiner.FrmMain.btnCombinePdfSharp_Click 14
PdfCombiner.FrmMain.cmbLanguage 10
PdfCombiner.FrmMain.cmbLanguage_SelectedIndexChanged 16
PdfCombiner.FrmMain.components 10
PdfCombiner.FrmMain.DeleteFilesFromListBox 16
PdfCombiner.FrmMain.Dispose 17
PdfCombiner.FrmMain.FormMembersNameInitialization 17
PdfCombiner.FrmMain.FrmMain 8
PdfCombiner.FrmMain.FrmMain_FormClosed 18
PdfCombiner.FrmMain.FrmMain_Shown 18
PdfCombiner.FrmMain.InitializeComponent 18
PdfCombiner.FrmMain.InitializeFileDialog 22
PdfCombiner.FrmMain.lbFiles 10
PdfCombiner.FrmMain.lbFiles_DragDrop 22
PdfCombiner.FrmMain.lbFiles_DragOver 23
PdfCombiner.FrmMain.lbFiles_KeyDown 23
PdfCombiner.FrmMain.lbFiles_MouseDown 23
PdfCombiner.FrmMain.lblDetails 10
PdfCombiner.FrmMain.menuItemDelete 10
PdfCombiner.FrmMain.menuItemDelete_Click 24
PdfCombiner.FrmMain.menuItemOrderByNameAscending 10
PdfCombiner.FrmMain.menuItemOrderByNameAscending_Click 24
PdfCombiner.FrmMain.menuItemOrderByNameDescending 10
PdfCombiner.FrmMain.menuItemOrderByNameDescending_Click 25
PdfCombiner.FrmMain.menuItemOrderByPathAscending 11
PdfCombiner.FrmMain.menuItemOrderByPathAscending_Click 25
PdfCombiner.FrmMain.menuItemOrderByPathDescending 11
PdfCombiner.FrmMain.menuItemOrderByPathDescending_Click 26
PdfCombiner.FrmMain.menuStrip 11
PdfCombiner.FrmMain.pbFiles 11
PdfCombiner.FrmMain.SortItemsByName 26
PdfCombiner.FrmMain.SortItemsByPath 27
PdfCombiner.Program 27
PdfCombiner.Program.Main 27
PdfCombiner.Properties 1
PdfCombiner.Properties namespace 1
    Classes 1
PdfCombiner.Properties.Resources 1
PdfCombiner.Properties.Resources.AppTitle 3
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.defaultInstance 4
PdfCombiner.sln 44
Program class 27
    about Program class 27
    Main 27
Program.cs 45

R

Resources class 1
    about Resources class 1
    AppTitle 3
    Culture 3
    resourceCulture 2
    resourceMan 3
    ResourceManager 3
    Resources 2
Resources.Designer.cs 45

S

Settings class 3

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.

about Settings class 3
defaultInstance 4
Settings.Designer.cs 47

Created with a commercial version of Doc-O-Matic. In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at support@toolsfactory.com.