PDF Combiner

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

Table of Contents

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
A	BaseException (see page 4)	This function is used to take exceptions which can be thrown in PDF Combiner App
S	ExceptionExtensions (see page 5)	This is class PdfCombiner.ExceptionExtensions.
	FrmMain (see page 6)	This is class PdfCombiner.FrmMain.
S	Program (see page 15)	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

C#

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Resources.Tools.StronglyType
dResourceBuilder",
"4.0.0.0")]
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
[global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]
internal class Resources;
```

File

Resources.Designer.cs (↗ see page 25)

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	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

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

1.1.1.1.1.2 Resources Fields

1.1.1.1.1.2.1 Resources.resourceCulture Field

C#

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

Description

This is resourceCulture, a member of class Resources.

1.1.1.1.1.2.2 Resources.resourceMan Field

C#

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

Description

This is resourceMan, a member of class Resources.

1.1.1.1.1.3 Resources Properties

1.1.1.1.1.3.1 Resources.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;
```

1.1.1.1.1.3.2 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

global::System.Configuration.ApplicationSettingsBase
PdfCombiner.Properties.Settings

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 26)

Description

This is class PdfCombiner.Properties.Settings.

Settings Fields

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

Settings Properties

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

1.1.1.1.2.1 Settings Fields

1.1.1.1.2.1.1 Settings.defaultInstance Field

C#

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

Description

This is defaultInstance, a member of class Settings.

1.1.1.1.2.2 Settings Properties

1.1.1.1.2.2.1 Settings.Default Property

C#

```
public static Settings Default;
```

Description

This is Default, a member of class Settings.

1.1.2 Classes

The following table lists classes in this documentation.

Classes

	Name	Description
A	BaseException (↗ see page 4)	This function is used to take exceptions which can be thrown in PDF Combiner App
S	ExceptionExtensions (↗ see page 5)	This is class PdfCombiner.ExceptionExtensions.
	FrmMain (↗ see page 6)	This is class PdfCombiner.FrmMain.
S	Program (↗ see page 15)	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

Exception
PdfCombiner.BaseException

C#

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

File

Exceptions.cs (↗ see page 17)

Methods

	Name	Description
	BaseException (↗ see page 5)	This is BaseException, a member of class BaseException.

1.1.2.1.1 BaseException Constructor

1.1.2.1.1.1 BaseException.BaseException Constructor ()

C#

```
protected BaseException();
```

Description

This is BaseException, a member of class BaseException.

Body Source

```
protected BaseException()
{
}
```

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

```
protected BaseException(string message) : base(message)
{
}
```

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

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

1.1.2.2 ExceptionExtensions Class

Class Hierarchy

C#

```
public static class ExceptionExtensions;
```

File

Exceptions.cs (see page 17)

Description

This is class PdfCombiner.ExceptionExtensions.

ExceptionExtensions Methods

	Name	Description
◆ S	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

```
public static string GetAllMessages(this Exception exception)
{
    var messages = exception.Message;
    if (exception.InnerException != null)
        messages += " " + exception.InnerException.GetAllMessages();
    if (!string.IsNullOrEmpty(exception.StackTrace))
        messages += " " + exception.StackTrace;
    if (!string.IsNullOrEmpty(exception.Source))
        messages += " " + exception.Source;
    return messages;
}
```

1.1.2.3 FrmMain Class

Class Hierarchy

Form
PdfCombiner.FrmMain

C#

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

File

FrmMain.Designer.cs (see page 22)

Description

This is class PdfCombiner.FrmMain.

Methods

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

FrmMain Fields

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

FrmMain Methods

	Name	Description
	btnAddFile_Click (↗ see page 9)	It is used to add single or multiple PDF files to combine When you choose file or files The listbox in the form will be filled with the name of the files Which you choose to combine
	btnAddFolder_Click (↗ see page 9)	This function is used to add PDF files in a folder which you choose in Folder Dialog recursively When you use it the added files will be seen in listbox
	btnClearList_Click (↗ see page 10)	This function is used to clear the file list in listbox
	btnCombine_Click (↗ see page 10)	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
	Dispose (↗ see page 12)	Clean up any resources being used.
	FrmMain_Shown (↗ see page 12)	This function is used to set some kind of parameters which can be used In PDF combination
	InitializeComponent (↗ see page 12)	Required method for Designer support - do not modify the contents of this method with the code editor.
S	InitializeFileDialog (↗ see page 14)	This function is used to set options of file dialog like Filter, start path etc.
	lbFiles_KeyDown (↗ see page 14)	This is lbFiles_KeyDown, a member of class FrmMain.

FrmMain Properties

	Name	Description
	AppFont (↗ see page 15)	This is AppFont, a member of class FrmMain.

1.1.2.3.1 FrmMain.FrmMain Constructor

C#

```
public FrmMain();
```

Description

This is FrmMain, a member of class FrmMain.

Body Source

```
public FrmMain()
{
    InitializeComponent();
}
```

1.1.2.3.2 FrmMain Fields

1.1.2.3.2.1 FrmMain.AppTitle Field

C#

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

Description

This is AppTitle, a member of class FrmMain.

1.1.2.3.2.2 FrmMain.btnAddFile Field

C#

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

Description

This is btnAddFile, a member of class FrmMain.

1.1.2.3.2.3 FrmMain.btnAddFolder Field

C#

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

Description

This is btnAddFolder, a member of class FrmMain.

1.1.2.3.2.4 FrmMain.btnClearList Field

C#

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

Description

This is btnClearList, a member of class FrmMain.

1.1.2.3.2.5 FrmMain.btnCombine Field

C#

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

Description

This is btnCombine, a member of class FrmMain.

1.1.2.3.2.6 FrmMain.components Field

Required designer variable.

C#

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

1.1.2.3.2.7 FrmMain.lbFiles Field

C#

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

Description

This is lbFiles, a member of class FrmMain.

1.1.2.3.2.8 FrmMain.lblDetails Field

C#

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

Description

This is lblDetails, a member of class FrmMain.

1.1.2.3.2.9 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	
EventArgs e	

Body Source

```
private void btnAddFile_Click(object sender, EventArgs e)
{
    using (var dialogAddFile = new OpenFileDialog())
    {
        InitializeFileDialog(dialogAddFile);
        var result = dialogAddFile.ShowDialog();
        if (result == DialogResult.OK && dialogAddFile.FileNames != null)
        {
            foreach (var file in dialogAddFile.FileNames)
            {
                if (!lbFiles.Items.Contains(file))
                    lbFiles.Items.Add(file);
            }
        }
    }
}
```

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	
EventArgs e	

Body Source

```
private void btnAddFolder_Click(object sender, EventArgs e)
{
    using (var dialogAddFolder = new FolderBrowserDialog())
    {
        var result = dialogAddFolder.ShowDialog();
        if (result == DialogResult.OK && !string.IsNullOrEmpty(dialogAddFolder.SelectedPath))
        {
            var fileNames = Directory.GetFiles(dialogAddFolder.SelectedPath, "*.pdf", SearchOption.AllDirectories);
            foreach (var file in fileNames)
            {
                if (!lbFiles.Items.Contains(file))
                    lbFiles.Items.Add(file);
            }
        }
    }
}
```

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	
EventArgs e	

Body Source

```
private void btnClearList_Click(object sender, EventArgs e)
{
    lbFiles.Items.Clear();
}
```

1.1.2.3.3.4 FrmMain.btnCombine_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

C#

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

Parameters

Parameters	Description
object sender	
EventArgs e	

Body Source

```
private void btnCombine_Click(object sender, EventArgs e)
{
    try
```

```
  {
    if (lbFiles.Items.Count < 2)
    {
      MessageBox.Show("There must be at least 2 PDF files to combine", AppTitle, MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
      using (var dialogExport = new FolderBrowserDialog())
      {
        var result = dialogExport.ShowDialog();
        if (result == DialogResult.OK && !string.IsNullOrEmpty(dialogExport.SelectedPath))
        {
          var outputFileName = dialogExport.SelectedPath + "/" + Guid.NewGuid().ToString() + ".pdf";
          var outputFile = new PdfDocument();
          outputFile.Options.CompressContentStreams = true;
          outputFile.Options.EnableCcittCompressionForBilevelImages = true;
          outputFile.Options.FlateEncodeMode = PdfFlateEncodeMode.BestCompression;
          var fileCount = lbFiles.Items.Count;
          var combinedFiles = 0;

          for (int i = 0; i < lbFiles.Items.Count; i++)
          {
            try
            {
              var inputDocument = PdfReader.Open(lbFiles.Items[i].ToString(), PdfDocumentOpenMode.Import);
              var count = inputDocument.PageCount;
              for (int idx = 0; idx < count; idx++)
              {
                var page = inputDocument.Pages[idx];
                outputFile.AddPage(page);
              }
              inputDocument.Close();
              combinedFiles++;
            }
            catch (Exception)
            {
              fileCount--;
            }

            pbFiles.Value = (combinedFiles * 100 / fileCount);
            if (pbFiles.Value > pbFiles.Maximum)
              pbFiles.Value = pbFiles.Maximum;
          }
          outputFile.Save(outputFileName);
          outputFile.Close();

          MessageBox.Show(fileCount + " PDF files successfully combined in " + outputFileName, AppTitle, MessageBoxButtons.OK, MessageBoxIcon.Information);

          pbFiles.Value = pbFiles.Minimum;
        }
      }
    }
  }
  catch (Exception ex)
  {
    MessageBox.Show("There is an error while combining PDF files in list. Details: " + ex.GetAllMessages(), AppTitle, MessageBoxButtons.OK, MessageBoxIcon.Error);
    pbFiles.Value = pbFiles.Minimum;
  }
}
```

1.1.2.3.3.5 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

```
protected override void Dispose(bool disposing)
{
    if (disposing && (components != null))
    {
        components.Dispose();
    }
    base.Dispose(disposing);
}
```

1.1.2.3.3.6 FrmMain.FrmMain_Shown Method

This function is used to set some kind of parameters which can be used In PDF combination

C#

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

Parameters

Parameters	Description
object sender	
EventArgs e	

Body Source

```
private void FrmMain_Shown(object sender, EventArgs e)
{
    //var options = new XPdfFontOptions(PdfFontEncoding.Unicode, PdfFontEmbedding.Always);
    //AppFont = new XFont("Times New Roman", 12, XFontStyle.Regular, options);
    //lblDetails.Text = "Font Type: " + AppFont.FontFamily.Name + ", Font Size: " + AppFont.Size + ", Font Style: " + AppFont.Style;
}
```

1.1.2.3.3.7 FrmMain.InitializeComponent Method

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

C#

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

Body Source

```
private void InitializeComponent()
{
    this.btnCombine = new System.Windows.Forms.Button();
    this.btnAddFolder = new System.Windows.Forms.Button();
    this.btnAddFile = new System.Windows.Forms.Button();
    this.btnClearList = new System.Windows.Forms.Button();
    this.lbFiles = new System.Windows.Forms.ListBox();
    this.lblDetails = new System.Windows.Forms.Label();
    this.pbFiles = new System.Windows.Forms.ProgressBar();
    this.SuspendLayout();
    //
    // btnCombine
    //
    this.btnCombine.Location = new System.Drawing.Point(544, 12);
    this.btnCombine.Name = "btnCombine";
    this.btnCombine.Size = new System.Drawing.Size(112, 23);
    this.btnCombine.TabIndex = 0;
    this.btnCombine.Text = "Combine Files";
    this.btnCombine.UseVisualStyleBackColor = true;
    this.btnCombine.Click += new System.EventHandler(this.btnCombine_Click);
    //
    // btnAddFolder
    //
    this.btnAddFolder.Location = new System.Drawing.Point(93, 12);
    this.btnAddFolder.Name = "btnAddFolder";
    this.btnAddFolder.Size = new System.Drawing.Size(75, 23);
    this.btnAddFolder.TabIndex = 2;
    this.btnAddFolder.Text = "Add Folder";
    this.btnAddFolder.UseVisualStyleBackColor = true;
    this.btnAddFolder.Click += new System.EventHandler(this.btnAddFolder_Click);
    //
    // btnAddFile
    //
    this.btnAddFile.Location = new System.Drawing.Point(12, 12);
    this.btnAddFile.Name = "btnAddFile";
    this.btnAddFile.Size = new System.Drawing.Size(75, 23);
    this.btnAddFile.TabIndex = 3;
    this.btnAddFile.Text = "Add Files";
    this.btnAddFile.UseVisualStyleBackColor = true;
    this.btnAddFile.Click += new System.EventHandler(this.btnAddFile_Click);
    //
    // btnClearList
    //
    this.btnClearList.Location = new System.Drawing.Point(174, 12);
    this.btnClearList.Name = "btnClearList";
    this.btnClearList.Size = new System.Drawing.Size(90, 23);
    this.btnClearList.TabIndex = 4;
    this.btnClearList.Text = "Clear File List";
    this.btnClearList.UseVisualStyleBackColor = true;
    this.btnClearList.Click += new System.EventHandler(this.btnClearList_Click);
    //
    // lbFiles
    //
    this.lbFiles.FormattingEnabled = true;
    this.lbFiles.Location = new System.Drawing.Point(12, 67);
    this.lbFiles.Name = "lbFiles";
    this.lbFiles.SelectionMode = System.Windows.Forms.SelectionMode.MultiExtended;
    this.lbFiles.Size = new System.Drawing.Size(644, 147);
    this.lbFiles.TabIndex = 5;
    this.lbFiles.KeyDown += new
    System.Windows.Forms.KeyEventHandler(this.lbFiles_KeyDown);
    //
    // lblDetails
    //
    this.lblDetails.AutoSize = true;
    this.lblDetails.Location = new System.Drawing.Point(12, 45);
    this.lblDetails.Name = "lblDetails";
    this.lblDetails.Size = new System.Drawing.Size(0, 13);
    this.lblDetails.TabIndex = 6;
    //
    // pbFiles
    //
    this.pbFiles.Location = new System.Drawing.Point(270, 12);
    this.pbFiles.Name = "pbFiles";
    this.pbFiles.Size = new System.Drawing.Size(268, 23);
    this.pbFiles.TabIndex = 7;
    //
    // FrmMain
```

```
    //
    this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
    this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
    this.ClientSize = new System.Drawing.Size(668, 227);
    this.Controls.Add(this.pbFiles);
    this.Controls.Add(this.lblDetails);
    this.Controls.Add(this.lbFiles);
    this.Controls.Add(this.btnClearList);
    this.Controls.Add(this.btnAddFile);
    this.Controls.Add(this.btnAddFolder);
    this.Controls.Add(this.btnCombine);
    this.Name = "FrmMain";
    this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
    this.Text = "PDF Combiner";
    this.FormClosed += new System.Windows.Forms.FormClosedEventHandler(this.FrmMain_FormClosed);
    this.Shown += new System.EventHandler(t
```

1.1.2.3.3.8 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

```
private static void InitializeFileDialog(OpenFileDialog dialogAddFile)
{
    dialogAddFile.Multiselect = true;
    dialogAddFile.Filter = "pdf files (*.pdf)|*.pdf";
    dialogAddFile.InitialDirectory = Application.StartupPath;
    dialogAddFile.Title = "Select PDF File/s";
    dialogAddFile.DefaultExt = "pdf";
}
```

1.1.2.3.3.9 FrmMain.lbFiles_KeyDown Method

C#

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

Description

This is lbFiles_KeyDown, a member of class FrmMain.

Body Source

```
/// <summary>
/// This function is used to delete files in listbox which are chosen
/// If you press the Delete button in your keyboard
/// The selected files will be deleted from the list
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void lbFiles_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyValue == Keys.Delete.GetHashCode())
    {
        var allItems = new List<string>();
        var removedItems = new List<string>();
        var fileCount = lbFiles.SelectedItems.Count;
        for (int i = 0; i < lbFiles.SelectedItems.Count; i++)
        {
            removedItems.Add(lbFiles.SelectedItems[i].ToString());
        }
        foreach (var item in lbFiles.Items)
        {
            if (!removedItems.Any(j => j == item.ToString()))
                allItems.Add(item.ToString());
        }

```

1.1.2.3.4 FrmMain Properties

1.1.2.3.4.1 FrmMain.AppFont Property

C#

```
public XFont AppFont;
```

Description

This is AppFont, a member of class FrmMain.

1.1.2.4 Program Class

Class Hierarchy

C#

```
public static class Program;
```

File

Program.cs (see page 24)

Description

This is class PdfCombiner.Program.

Program Methods

	Name	Description
S	Main (see page 15)	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

```
[STAThread]
static void Main()
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    Application.Run(new FrmMain());
}
```

1.2 Files

The following table lists files in this documentation.

Files

Name	Description
AssemblyInfo.cs (↗ see page 16)	This is file AssemblyInfo.cs.
Exceptions.cs (↗ see page 17)	This is file Exceptions.cs.
FrmMain.cs (↗ see page 18)	This is file FrmMain.cs.
FrmMain.Designer.cs (↗ see page 22)	This is file FrmMain.Designer.cs.
PdfCombiner.csproj (↗ see page 24)	This is file PdfCombiner.csproj.
PdfCombiner.sln (↗ see page 24)	This is file PdfCombiner.sln.
Program.cs (↗ see page 24)	This is file Program.cs.
Resources.Designer.cs (↗ see page 25)	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 26)	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

```
?using System.Reflection;
using System.Runtime.CompilerServices;
using System.Runtime.InteropServices;

// General Information about an assembly is controlled through the following
// set of attributes. Change these attribute values to modify the information
// associated with an assembly.
[assembly: AssemblyTitle("PdfCombiner")]
[assembly: AssemblyDescription("")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyCompany("")]
[assembly: AssemblyProduct("PdfCombiner")]
[assembly: AssemblyCopyright("Copyright ©  2022")]
[assembly: AssemblyTrademark("")]
[assembly: AssemblyCulture("")]

// Setting ComVisible to false makes the types in this assembly not visible
// to COM components.  If you need to access a type in this assembly from
// COM, set the ComVisible attribute to true on that type.
[assembly: ComVisible(false)]

// The following GUID is for the ID of the typelib if this project is exposed to COM
[assembly: Guid("48e34ee8-dfaa-48db-acfd-70573cee9c00")]

// Version information for an assembly consists of the following four values:
//
//      Major Version
//      Minor Version
//      Build Number
//      Revision
//
// You can specify all the values or you can default the Build and Revision Numbers
// by using the '*' as shown below:
// [assembly: AssemblyVersion("1.0.*")]
[assembly: AssemblyVersion("1.0.0.0")]
[assembly: AssemblyFileVersion("1.0.0.0")]
```

1.2.2 Exceptions.cs

This is file Exceptions.cs.

Body Source

```
?using System;

namespace PdfCombiner
{
    /// <summary>
    /// This function is used to take exceptions which can be thrown in
    /// PDF Combiner App
    /// </summary>
    [Serializable]
    public abstract class BaseException : Exception
    {
        protected BaseException()
        {
        }

        protected BaseException(string message) : base(message)
        {
        }

        protected BaseException(string message, Exception innerException) : base(message, innerException)
        {
        }
    }

    public static class ExceptionExtensions
    {
        /// <summary>
        /// This method is used to take all messages with stack trace info in exception
        /// </summary>
        /// <param name="exception">Exception</param>
        /// <returns>Detailed Error Message</returns>
        public static string GetAllMessages(this Exception exception)
        {
            var messages = exception.Message;
            if (exception.InnerException != null)
                messages += " " + exception.InnerException.GetAllMessages();
            if (!string.IsNullOrEmpty(exception.StackTrace))
                messages += " " + exception.StackTrace;
            if (!string.IsNullOrEmpty(exception.Source))
                messages += " " + exception.Source;
            return messages;
        }
    }
}
```

Namespaces

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

1.2.3 FrmMain.cs

This is file FrmMain.cs.

Body Source

```
?using PdfSharp.Drawing;
using PdfSharp.Pdf;
using PdfSharp.Pdf.IO;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Windows.Forms;

namespace PdfCombiner
{
    public partial class FrmMain : Form
    {
        public XFont AppFont { get; set; }
        public const string AppTitle = "PDF Combiner";

        public FrmMain()
        {
            InitializeComponent();
        }

        /// <summary>
        /// This function is used to set some kind of parameters which can be used
        /// In PDF combination
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void FrmMain_Shown(object sender, EventArgs e)
        {
            //var options = new XPdfFontOptions(PdfFontEncoding.Unicode, PdfFontEmbedding.Always);
            //AppFont = new XFont("Times New Roman", 12, XFontStyle.Regular, options);
            //lblDetails.Text = "Font Type:  " + AppFont.FontFamily.Name + ", Font Size:  " + AppFont.Size + ", Font Style:  " + AppFont.Style;
        }

        /// <summary>
        /// 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
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void btnAddFile_Click(object sender, EventArgs e)
        {
            using (var dialogAddFile = new OpenFileDialog())
            {
                InitializeFileDialog(dialogAddFile);
                var result = dialogAddFile.ShowDialog();
                if (result == DialogResult.OK && dialogAddFile.FileNames != null)
                {
                    foreach (var file in dialogAddFile.FileNames)
                    {
                        if (!lbFiles.Items.Contains(file))
                            lbFiles.Items.Add(file);
                    }
                }
            }
        }

        /// <summary>
        /// This function is used to set options of file dialog like
        /// Filter, start path etc.
        /// </summary>
        /// <param name="dialogAddFile">File Dialog</param>
        private static void InitializeFileDialog(OpenFileDialog dialogAddFile)
        {
            dialogAddFile.Multiselect = true;
            dialogAddFile.Filter = "pdf files (*.pdf)|*.pdf";
            dialogAddFile.InitialDirectory = Application.StartupPath;
            dialogAddFile.Title = "Select PDF File/s";
            dialogAddFile.DefaultExt = "pdf";
        }

        /// <summary>
        /// This function is used to clear the file list in listbox
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void btnClearList_Click(object sender, EventArgs e)
        {
            lbFiles.Items.Clear();
        }

        /// <summary>
        /// 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
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void btnAddFolder_Click(object sender, EventArgs e)
        {
            using (var dialogAddFolder = new FolderBrowserDialog())
            {
                var result = dialogAddFolder.ShowDialog();
                if (result == DialogResult.OK && !string.IsNullOrEmpty(dialogAddFolder.SelectedPath))
                {
                    var fileNames = Directory.GetFiles(dialogAddFolder.SelectedPath, "*.pdf", SearchOption.AllDirectories);
                    foreach (var file in fileNames)
                    {
                        if (!lbFiles.Items.Contains(file))
                            lbFiles.Items.Add(file);
                    }
                }
            }
        }

        //private void btnSetFont_Click(object sender, EventArgs e)
        //{
        //   using (var dialogSetFont = new FontDialog())
        //   {
        //       var result = dialogSetFont.ShowDialog();
        //       if (result == DialogResult.OK && dialogSetFont.Font != null)
        //       {
        //           var fontDialog = dialogSetFont.Font;
        //           var options = new XPdfFontOptions(PdfFontEncoding.Unicode, PdfFontEmbedding.Always);
        //           AppFont = new XFont(fontDialog.FontFamily.Name, fontDialog.Size, (XFontStyle)fontDialog.Style, options);
        //           lblDetails.Text = "Font Type: " + AppFont.FontFamily.Name + ", Font Size: " + AppFont.Size + ", Font Style: " + AppFont.Style;
        //       }
        //   }
        //}

        /// <summary>
        /// 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
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void btnCombine_Click(object sender, EventArgs e)
        {
            try
            {
                if (lbFiles.Items.Count < 2)
                {
                    MessageBox.Show("There must be at least 2 PDF files to combine", AppTitle, MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
                else
                {
                    using (var dialogExport = new FolderBrowserDialog())
                    {
                        var result = dialogExport.ShowDialog();
                        if (result == DialogResult.OK && !string.IsNullOrEmpty(dialogExport.SelectedPath))
                        {
                            var outputFileName = dialogExport.SelectedPath + "/" + Guid.NewGuid().ToString() + ".pdf";
                            var outputFile = new PdfDocument();
                            outputFile.Options.CompressContentStreams = true;
                            outputFile.Options.EnableCcittCompressionForBilevelImages = true;
                            outputFile.Options.FlateEncodeMode = PdfFlateEncodeMode.BestCompression;
                            var fileCount = lbFiles.Items.Count;
                            var combinedFiles = 0;

                            for (int i = 0; i < lbFiles.Items.Count; i++)
                            {
                                try
                                {
                                    var inputDocument = PdfReader.Open(lbFiles.Items[i].ToString(), PdfDocumentOpenMode.Import);
                                    var count = inputDocument.PageCount;
                                    for (int idx = 0; idx < count; idx++)
                                    {
                                        var page = inputDocument.Pages[idx];
                                        outputFile.AddPage(page);
                                    }
                                    inputDocument.Close();
                                    combinedFiles++;
                                }
                                catch (Exception)
                                {
                                    fileCount--;
                                }

                                pbFiles.Value = (combinedFiles * 100 / fileCount);
                                if (pbFiles.Value > pbFiles.Maximum)
                                    pbFiles.Value = pbFiles.Maximum;
                            }
                            outputFile.Save(outputFileName);
                            outputFile.Close();

                            MessageBox.Show(fileCount + " PDF files successfully combined in " + outputFileName, AppTitle, MessageBoxButtons.OK, MessageBoxIcon.Information);

                            pbFiles.Value = pbFiles.Minimum;
                        }
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show("There is an error while combining PDF files in list. Details: " + ex.GetAllMessages(), AppTitle, MessageBoxButtons.OK, MessageBoxIcon.Error);
                pbFiles.Value = pbFiles.Minimum;
            }
        }

        /// <summary>
        /// This function is used to delete files in listbox which are chosen
        /// If you press the Delete button in your keyboard
        /// The selected files will be deleted from the list
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void lbFiles_KeyDown(object sender, KeyEventArgs e)
        {
            if (e.KeyValue == Keys.Delete.GetHashCode())
            {
                var allItems = new List<string>();
                var removedItems = new List<string>();
                var fileCount = lbFiles.SelectedItems.Count;
                for (int i = 0; i < lbFiles.SelectedItems.Count; i++)
                {
                    removedItems.Add(lbFiles.SelectedItems[i].ToString());
                }
                foreach (var item in lbFiles.Items)
                {
                    if (!removedItems.Any(j => j == item.ToString()))
                        allItems.Add(item.ToString());
                }
                lbFiles.Items.Clear();
                foreach (var item in allItems)
                {
                    lbFiles.Items.Add(item);
                }
                MessageBox.Show(fileCount + " files are deleted from the list.", AppTitle, MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
        }

        /// <summary>
        /// This function is used to terminate application
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void FrmMain_FormClosed(object sender, FormClosedEventArgs e)
        {
            MessageBox.Show("Thanks for using this app", AppTitle, MessageBoxButtons.OK, MessageBoxIcon.Information);
            Application.Exit();
        }
    }
}
```

Namespaces

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

1.2.4 FrmMain.Designer.cs

This is file FrmMain.Designer.cs.

Body Source

```
?namespace PdfCombiner
{
    partial class FrmMain
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.btnCombine = new System.Windows.Forms.Button();
            this.btnAddFolder = new System.Windows.Forms.Button();
            this.btnAddFile = new System.Windows.Forms.Button();
            this.btnClearList = new System.Windows.Forms.Button();
            this.lbFiles = new System.Windows.Forms.ListBox();
            this.lblDetails = new System.Windows.Forms.Label();
            this.pbFiles = new System.Windows.Forms.ProgressBar();
            this.SuspendLayout();
            //
            // btnCombine
            //
            this.btnCombine.Location = new System.Drawing.Point(544, 12);
            this.btnCombine.Name = "btnCombine";
            this.btnCombine.Size = new System.Drawing.Size(112, 23);
            this.btnCombine.TabIndex = 0;
            this.btnCombine.Text = "Combine Files";
            this.btnCombine.UseVisualStyleBackColor = true;
            this.btnCombine.Click += new System.EventHandler(this.btnCombine_Click);
            //
            // btnAddFolder
            //
            this.btnAddFolder.Location = new System.Drawing.Point(93, 12);
            this.btnAddFolder.Name = "btnAddFolder";
            this.btnAddFolder.Size = new System.Drawing.Size(75, 23);
            this.btnAddFolder.TabIndex = 2;
            this.btnAddFolder.Text = "Add Folder";
            this.btnAddFolder.UseVisualStyleBackColor = true;
            this.btnAddFolder.Click += new
```

```
System.EventHandler(this.btnAddFolder_Click);
            //
            // btnAddFile
            //
            this.btnAddFile.Location = new System.Drawing.Point(12, 12);
            this.btnAddFile.Name = "btnAddFile";
            this.btnAddFile.Size = new System.Drawing.Size(75, 23);
            this.btnAddFile.TabIndex = 3;
            this.btnAddFile.Text = "Add Files";
            this.btnAddFile.UseVisualStyleBackColor = true;
            this.btnAddFile.Click += new System.EventHandler(this.btnAddFile_Click);
            //
            // btnClearList
            //
            this.btnClearList.Location = new System.Drawing.Point(174, 12);
            this.btnClearList.Name = "btnClearList";
            this.btnClearList.Size = new System.Drawing.Size(90, 23);
            this.btnClearList.TabIndex = 4;
            this.btnClearList.Text = "Clear File List";
            this.btnClearList.UseVisualStyleBackColor = true;
            this.btnClearList.Click += new
System.EventHandler(this.btnClearList_Click);
            //
            // lbFiles
            //
            this.lbFiles.FormattingEnabled = true;
            this.lbFiles.Location = new System.Drawing.Point(12, 67);
            this.lbFiles.Name = "lbFiles";
            this.lbFiles.SelectionMode =
System.Windows.Forms.SelectionMode.MultiExtended;
            this.lbFiles.Size = new System.Drawing.Size(644, 147);
            this.lbFiles.TabIndex = 5;
            this.lbFiles.KeyDown += new
System.Windows.Forms.KeyEventHandler(this.lbFiles_KeyDown);
            //
            // lblDetails
            //
            this.lblDetails.AutoSize = true;
            this.lblDetails.Location = new System.Drawing.Point(12, 45);
            this.lblDetails.Name = "lblDetails";
            this.lblDetails.Size = new System.Drawing.Size(0, 13);
            this.lblDetails.TabIndex = 6;
            //
            // pbFiles
            //
            this.pbFiles.Location = new System.Drawing.Point(270, 12);
            this.pbFiles.Name = "pbFiles";
            this.pbFiles.Size = new System.Drawing.Size(268, 23);
            this.pbFiles.TabIndex = 7;
            //
            // FrmMain
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(668, 227);
            this.Controls.Add(this.pbFiles);
            this.Controls.Add(this.lblDetails);
            this.Controls.Add(this.lbFiles);
            this.Controls.Add(this.btnClearList);
            this.Controls.Add(this.btnAddFile);
            this.Controls.Add(this.btnAddFolder);
            this.Controls.Add(this.btnCombine);
            this.Name = "FrmMain";
            this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
            this.Text = "PDF Combiner";
            this.FormClosed += new
System.Windows.Forms.FormClosedEventHandler(this.FrmMain_FormClosed);
            this.Shown += new System.EventHandler(this.FrmMain_Shown);
            this.ResumeLayout(false);
            this.PerformLayout();

        }

        #endregion

        private System.Windows.Forms.Button btnCombine;
        private System.Windows.Forms.Button btnAddFolder;
        private System.Windows.Forms.Button btnAddFile;
        private System.Windows.Forms.Button btnClearList;
        private System.Windows.Forms.ListBox lbFiles;
        private System.Windows.Forms.Label lblDetails;
        private System.Windows.Forms.ProgressBar pbFiles;
    }
}

```

Namespaces

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

1.2.5 PdfCombiner.csproj

This is file PdfCombiner.csproj.

1.2.6 PdfCombiner.sln

This is file PdfCombiner.sln.

1.2.7 Program.cs

This is file Program.cs.

Body Source

```
?using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;

namespace PdfCombiner
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new FrmMain());
        }
    }
}
```

Namespaces

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

1.2.8 Resources.Designer.cs

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

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

Body Source

```
?//------------------------------------------------------------------------------
// <auto-generated>
//     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.
// </auto-generated>
//------------------------------------------------------------------------------

namespace PdfCombiner.Properties
{


    /// <summary>
    ///   A strongly-typed resource class, for looking up localized strings, etc.
    /// </summary>
    // 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.
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Resources.Tools.StronglyTypedResourceBuilder", "4.0.0.0")]
    [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
    [global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]
    internal class Resources
    {

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

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

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

        /// <summary>
        ///   Returns the cached ResourceManager instance used by this class.
        /// </summary>
[global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableState.Advanced)]
        internal static global::System.Resources.ResourceManager ResourceManager
        {
            get
            {
                if ((resourceMan == null))
                {
                    global::System.Resources.ResourceManager temp = new
global::System.Resources.ResourceManager("PdfCombiner.Properties.Resources",
typeof(Resources).Assembly);
                    resourceMan = temp;
                }
                return resourceMan;
            }
        }

        /// <summary>
        ///   Overrides the current thread's CurrentUICulture property for all
        ///   resource lookups using this strongly typed resource class.
        /// </summary>
[global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.Editor
BrowsableState.Advanced)]
        internal static global::System.Globalization.CultureInfo Culture
        {
            get
            {
                return resourceCulture;
            }
            set
            {
                resourceCulture = value;
            }
        }
    }
}
```

Namespaces

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

1.2.9 Settings.Designer.cs

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

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

Body Source

```
?//------------------------------------------------------------------------------
// <auto-generated>
//     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.
// </auto-generated>
//------------------------------------------------------------------------------

namespace PdfCombiner.Properties
{


    [global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.VisualStudio.Editors.Sett
ingsDesigner.SettingsSingleFileGenerator",
"11.0.0.0")]
```

```
    internal sealed partial class Settings : global::System.Configuration.ApplicationSettingsBase
    {

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

        public static Settings Default
        {
            get
            {
                return defaultInstance;
            }
        }
    }
}
```

Namespaces

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

Index