AllPageElements.aspx

```
<%@ Page language="c#" Inherits="Web_CSharp.GeneratorExamples.AllPageElements" CodeFile="AllPageElements.aspx.cs" %>
```

```
using System;
using System.Text;
using System.IO;
using System.Security.Cryptography;
using ceTe.DynamicPDF;
using ceTe.DynamicPDF.IO;
using ceTe.DynamicPDF.Text;
using ceTe.DynamicPDF.PageElements;
using ceTe.DynamicPDF.PageElements.Forms;
using ceTe.DynamicPDF.PageElements.BarCoding;
using ceTe.DynamicPDF.PageElements.Charting;
using ceTe.DynamicPDF.PageElements.Charting.Axes;
using ceTe.DynamicPDF.PageElements.Charting.Series;
using ceTe.DynamicPDF.PageElements.Charting.Values;
using ceTe.DynamicPDF.PageElements.Html;

namespace Web_CSharp.GeneratorExamples
{
    public partial class AllPageElements : System.Web.UI.Page
    {
        #region Member Variables
        private int pageCounter = 0;
        #endregion

        #region StopSign Custom Page Element Class
        /// <summary>
        /// Custom page element
        /// </summary>
        public class StopSign : ceTe.DynamicPDF.PageElement
        {
            #region Member Variables
            private float x = 0.0f;
            private float y = 0.0f;
            private float w = 0.0f;
            private float h = 0.0f;
            #endregion

            #region Constructors
            /// <summary>
            /// Initilizes a new instance of the <see cref="MyPageElement"/> class.
            /// </summary>
            /// <param name="x">X coordinate of the page element.</param>
            /// <param name="y">Y coordinate of the page element.</param>
            /// <param name="width">Width of the page element.</param>
            /// <param name="height">Height of the page element.</param>
            public StopSign(float x, float y, float width, float height)
            {
                this.x = x;
                this.y = y;
                this.w = width;
```

```
51                 this.h = height;
52             }
53             #endregion
54
55             #region Public methods
56             /// <summary>
57             /// Overriden. Draws a MyPageElement page element to the page.
58             /// </summary>
59             /// <param name="writer"><see cref="PageWriter"/> object to recieve the PDF output.</param>
60             public override void Draw(PageWriter writer)
61             {
62                 float v = h / (float)3.414;
63
64                 // End text mode
65                 writer.Write_ET();
66                 writer.Write_m_(x + v, y);
67                 writer.Write_RG(RgbColor.White);
68                 writer.Write_rg_(RgbColor.Red);
69
70                 // Draws the axis lines to the page
71                 writer.Write_l_(x + h - v, y);
72                 writer.Write_l_(x + h, y + v);
73                 writer.Write_l_(x + h, y + h - v);
74                 writer.Write_l_(x + h - v, y + h);
75                 writer.Write_l_(x + v, y + h);
76                 writer.Write_l_(x, y + h - v);
77                 writer.Write_l_(x, y + v);
78                 writer.Write_b_();
79                 char[] text = new char[] { 'S', 'T', 'O', 'P' };
80
81                 // Begin text mode
82                 writer.Write_BT();
83                 writer.SetFont(ceTe.DynamicPDF.Font.Helvetica, 16);
84                 writer.Write_rg_(RgbColor.White);
85                 writer.Write_Tm((float)88.5, y + (float)(w / 1.65));
86
87                 // Wirte the text
88                 writer.Write_Tj_(text, false);
89
90                 // End text mode
91                 writer.Write_ET();
92             }
93             #endregion
94         }
95         #endregion
96
97         #region Page_Load Method
98         protected void Page_Load(object sender, System.EventArgs e)
99         {
100            // Create a document and set it's properties
```

```
            ceTe.DynamicPDF.Document document = new ceTe.DynamicPDF.Document();
            document.Creator = "AllPageElements.aspx";
            document.Author = "ceTe Software";
            document.Title = "All Page Elements";

            // Add the top level outline to the document
            Outline parentOutline = document.Outlines.Add("Top Level Outline", new XYDestination(1, 0, 0));
            parentOutline.Color = new WebColor("0000FF");
            parentOutline.Style = TextStyle.Bold;

            // Add pages to the document
            AddPageElementPage(document, parentOutline);
            AddTableAndOther(document, parentOutline);
            AddChartsPage(document, parentOutline);
            AddFormElementPage(document, parentOutline);
            AddUPCEANPage(document, parentOutline);
            AddPostalBarCodePage(document, parentOutline);
            AddIsbnIssnIsmnPage(document, parentOutline);
            AddBarCodePage(document, parentOutline);
            Add2DBarCodePage(document, parentOutline);
            AddWatermark(document, parentOutline);

            // Outputs the document to the current web page
            document.DrawToWeb("AllPageElements.pdf");
        }


        #endregion

        #region Private Methods

        private void AddIsbn(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "ISBN Barcode", x, y, 204, 99);
            Isbn barCode = new Isbn("978-1-23-456789-7", x, y + 21);
            barCode.ShowText = true;
            barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
            elements.Add(barCode);
        }

        private void AddIsbnSup2(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "ISBN Barcode, 2 digit supplement", x, y, 204, 99);
            IsbnSup2 barCode = new IsbnSup2("978-1-23-456789-7", "12", x, y + 21);
            barCode.ShowText = true;
            barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
            elements.Add(barCode);
        }

        private void AddIsbnSup5(Group elements, float x, float y)
```

```
        {
            AddCaptionAndRectangle(elements, "ISBN Barcode, 5 digit supplement", x, y, 204, 99);
            IsbnSup5 barCode = new IsbnSup5("978-1-23-456789-7", "12345", x, y + 21);
            barCode.ShowText = true;
            barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
            elements.Add(barCode);
        }

        private void AddIsmn(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "ISMN Barcode", x, y, 204, 99);
            Ismn barCode = new Ismn("979-0-1234-5678", x, y + 21);
            barCode.ShowText = true;
            barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
            elements.Add(barCode);
        }

        private void AddIsmnSup2(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "ISMN Barcode, 2 digit supplement", x, y, 204, 99);
            IsmnSup2 barCode = new IsmnSup2("979-0-1234-5678", "12", x, y + 21);
            barCode.ShowText = true;
            barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
            elements.Add(barCode);
        }

        private void AddIsmnSup5(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "ISMN Barcode, 5 digit supplement", x, y, 204, 99);
            IsmnSup5 barCode = new IsmnSup5("979-0-1234-5678", "12345", x, y + 21);
            barCode.ShowText = true;
            barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
            elements.Add(barCode);
        }

        private void AddIssn(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "ISSN Barcode", x, y, 204, 99);
            Issn barCode = new Issn("977-1234-56700", x, y + 21);
            barCode.ShowText = true;
            barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
            elements.Add(barCode);
        }

        private void AddIssnSup2(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "ISSN Barcode, 2 digit supplement", x, y, 204, 99);
            IssnSup2 barCode = new IssnSup2("977-1234-56700", "12", x, y + 21);
            barCode.ShowText = true;
            barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
```

```
            elements.Add(barCode);
        }

        private void AddIssnSup5(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "ISSN Barcode, 5 digit supplement", x, y, 204, 99);
            IssnSup5 barCode = new IssnSup5("977-1234-56700", "12345", x, y + 21);
            barCode.ShowText = true;
            barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
            elements.Add(barCode);
        }

        private void AddSingaporePost(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "Singapore Post Barcode", x, y, 225, 80);
            BarCode barCode = new SingaporePost("208154", x, y + 15, true);
            barCode.Height = 35;
            barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
            barCode.Y += (65 - barCode.Height) / 2;
            elements.Add(barCode);
        }

        private void AddKix(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "Kix (Royal TNT Post) Barcode", x, y, 225, 80);
            BarCode barCode = new Kix("1231FZ13XHS", x, y + 15, true);
            barCode.Height = 35;
            barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
            barCode.Y += (65 - barCode.Height) / 2;
            elements.Add(barCode);
        }

        private void AddDeutschePostLeitcode(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "Deutsche Post Leitcode", x, y, 225, 80);
            BarCode barCode = new DeutschePostLeitcode("1234567890123", x, y + 15, 48);
            barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
            barCode.Y += (65 - barCode.Height) / 2;
            elements.Add(barCode);
        }

        private void AddDeutschePostIdentcode(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "Deutsche Post Identcode", x, y, 225, 80);
            BarCode barCode = new DeutschePostIdentcode("12345678901", x, y + 15, 48);
            barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
            barCode.Y += (65 - barCode.Height) / 2;
            elements.Add(barCode);
        }

```

```
        private void AddAustraliaPost(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "Australia Post Barcode", x, y, 225, 80);
            BarCode barCode = new AustraliaPost("1139549554", x, y + 15, true);
            barCode.Height = 35;
            barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
            barCode.Y += (65 - barCode.Height) / 2;
            elements.Add(barCode);
        }

        private void AddAztec(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "Aztec Barcode", x, y, 225, 100);
            Aztec aztec = new Aztec("Aztec Barcode", x, y + 15, AztecSymbolSize.Auto);
            aztec.XDimension = 2.5f;
            aztec.X += (225 - aztec.GetSymbolWidth()) / 2;
            aztec.Y += (85 - aztec.GetSymbolHeight()) / 2;
            elements.Add(aztec);
        }

        private void AddStackedGs1Expanded(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "Stacked GS1 DataBar Expanded Barcode", x, y, 225, 100);
            StackedGS1DataBar barCode = new StackedGS1DataBar("(01)1234567890123", x, y + 15, 25,
StackedGS1DataBarType.ExpandedStacked);
            barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
            barCode.Y += (85 - barCode.GetSymbolHeight()) / 2;
            elements.Add(barCode);
        }

        private void AddStackedGs1Stacked(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "Stacked GS1 DataBar Barcode", x, y, 225, 100);
            StackedGS1DataBar barCode = new StackedGS1DataBar("(01)1234567890123", x, y + 15, 25,
StackedGS1DataBarType.Stacked);
            barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
            barCode.Y += (85 - barCode.GetSymbolHeight()) / 2;
            elements.Add(barCode);
        }

        private void AddStackedGs1Omni(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "Stacked GS1 DataBar Omnidirectional Barcode", x, y, 225, 100);
            StackedGS1DataBar barCode = new StackedGS1DataBar("(01)1234567890123", x, y + 15, 25,
StackedGS1DataBarType.StackedOmnidirectional);
            barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
            barCode.Y += (85 - barCode.GetSymbolHeight()) / 2;
            elements.Add(barCode);
        }

```

```
        private void AddGs1DataBarExpanded(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "GS1 DataBar Expanded Barcode", x, y, 225, 80);
            BarCode barCode = new GS1DataBar("(01)1234567890123", x, y + 15, 48, GS1DataBarType.Expanded);
            barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
            barCode.Y += (65 - barCode.Height) / 2;
            elements.Add(barCode);
        }

        private void AddGs1DataBarLimited(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "GS1 DataBar Limited Barcode", x, y, 225, 80);
            BarCode barCode = new GS1DataBar("(01)1234567890123", x, y + 15, 48, GS1DataBarType.Limited);
            barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
            barCode.Y += (65 - barCode.Height) / 2;
            elements.Add(barCode);
        }

        private void AddGs1DataBarOmnidirectional(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "GS1 DataBar Omnidirectional Barcode", x, y, 225, 80);
            BarCode barCode = new GS1DataBar("(01)9889876543210", x, y + 15, 48, GS1DataBarType.Omnidirectional);
            barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
            barCode.Y += (65 - barCode.Height) / 2;
            elements.Add(barCode);
        }

        private void AddEan14(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "EAN-14 Barcode", x, y, 225, 80);
            BarCode barCode = new Ean14("1234567890123", x, y + 15, 48);
            barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
            barCode.Y += (65 - barCode.Height) / 2;
            elements.Add(barCode);
        }

        private void AddItf14(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "ITF-14 Barcode", x, y, 225, 80);
            BarCode barCode = new Itf14("1234567890123", x, y + 15, 48);
            barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
            barCode.Y += (65 - barCode.Height) / 2;
            elements.Add(barCode);
        }

        private void AddIata25(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "IATA-25 Barcode", x, y, 225, 80);
            BarCode barCode = new Iata25("1234567", x, y + 15, 48);
            barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
```

```
            barCode.Y += (65 - barCode.Height) / 2;
            elements.Add(barCode);
        }

        private void AddExtendedCode93(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "Extended Code 93 Barcode", x, y, 225, 80);
            BarCode barCode = new Code93("Code 93", x, y + 15, 48);
            barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
            barCode.Y += (65 - barCode.Height) / 2;
            elements.Add(barCode);
        }

        private void AddCode93(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "Code 93 Barcode", x, y, 225, 80);
            BarCode barCode = new Code93("CODE 93", x, y + 15, 48);
            barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
            barCode.Y += (65 - barCode.Height) / 2;
            elements.Add(barCode);
        }

        private void AddExtendedCode39(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "Extended Code 3 of 9 Barcode", x, y, 225, 80);
            BarCode barCode = new Code39("Code 39", x, y + 15, 48);
            barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
            barCode.Y += (65 - barCode.Height) / 2;
            elements.Add(barCode);
        }

        private void AddCode11(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "Code 11 Barcode", x, y, 225, 80);
            BarCode barCode = new Code11("12345678", x, y + 15, 48);
            barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
            barCode.Y += (65 - barCode.Height) / 2;
            elements.Add(barCode);
        }

        private void AddMSICode(Group elements, int x, int y)
        {
            AddCaptionAndRectangle(elements, "MSI Barcode", x, y, 225, 80);
            MsiBarcode msi = new MsiBarcode("1234567890", x, y, 50, true);
            msi.Y += (80 - msi.Height) / 2 + 8;
            msi.X += (225 - msi.GetSymbolWidth()) / 2;
            elements.Add(msi);
        }
        private void AddRm4sccCode(Group elements, int x, int y)
        {
```

```
            AddCaptionAndRectangle(elements, "RM4SCC Barcode", x, y, 225, 80);
            Rm4scc rm4sccCode = new Rm4scc("1457826RA", x, y, true);
            rm4sccCode.Height = 35;
            rm4sccCode.Y += (65 - rm4sccCode.Height) / 2 + 13;
            rm4sccCode.X += (225 - rm4sccCode.GetSymbolWidth()) / 2;
            elements.Add(rm4sccCode);
        }
        private void AddBarCodePage(Document document, Outline parentOutline)
        {
            // Increments a page counter
            pageCounter++;

            // Adds to outlines to the page
            parentOutline.ChildOutlines.Add("Page " + pageCounter + " - Other Linear Barcodes", new ZoomDestination(pageCounter, PageZoom.FitPage));

            // Create a page to add to the document
            ceTe.DynamicPDF.Page page = new ceTe.DynamicPDF.Page(PageSize.Letter, PageOrientation.Portrait, 54);

            // Uncomment the line below to add a layout grid to the page
            //page.Elements.Add( new LayoutGrid() );

            // Add the page elements to the page
            AddCodabar(page.Elements, 0, 0);
            AddCode11(page.Elements, 0, 88);
            AddCode128(page.Elements, 0, 176);
            AddCode25(page.Elements, 0, 264);
            AddCode39(page.Elements, 0, 352);
            AddExtendedCode39(page.Elements, 0, 440);
            AddCode93(page.Elements, 0, 528);
            AddExtendedCode93(page.Elements, 0, 616);

            AddIata25(page.Elements, 279, 0);
            AddInterleaved25(page.Elements, 279, 88);
            AddItf14(page.Elements, 279, 176);
            AddMSICode(page.Elements, 279, 264);
            AddEan14(page.Elements, 279, 352);
            AddGs1DataBarOmnidirectional(page.Elements, 279, 440);
            AddGs1DataBarLimited(page.Elements, 279, 528);
            AddGs1DataBarExpanded(page.Elements, 279, 616);

            document.Pages.Add(page);
        }
        private void Add2DBarCodePage(Document document, Outline parentOutline)
        {
            // Increments a page counter
            pageCounter++;

            // Adds to outlines to the page
```

```
            parentOutline.ChildOutlines.Add("Page " + pageCounter + " - 2D and Stacked GS1 Databar Barcodes", new

            // Create a page to add to the document
            ceTe.DynamicPDF.Page page = new ceTe.DynamicPDF.Page(PageSize.Letter, PageOrientation.Portrait, 54);

            // Uncomment the line below to add a layout grid to the page
            //page.Elements.Add( new LayoutGrid() );

            // Add the page elements to the page
            AddStackedGs1Omni(page.Elements, 0, 0);
            AddStackedGs1Stacked(page.Elements, 0, 108);
            AddStackedGs1Expanded(page.Elements, 0, 216);
            AddAztec(page.Elements, 0, 324);
            AddQrCode(page.Elements, 0, 432);

            AddPdf417(page.Elements, 279, 0);
            AddPdf417Macro(page.Elements, 279, 108);
            AddDataMatrixBarcode(page.Elements, 279, 324);

            document.Pages.Add(page);
        }
        private void AddPostalBarCodePage(Document document, Outline parentOutline)
        {
            // Increments a page counter
            pageCounter++;

            // Adds to outlines to the page
            parentOutline.ChildOutlines.Add("Page " + pageCounter + " - Postal Barcodes", new ZoomDestination(pageCounter,
PageZoom.FitPage));

            // Create a page to add to the document
            ceTe.DynamicPDF.Page page = new ceTe.DynamicPDF.Page(PageSize.Letter, PageOrientation.Portrait, 54);

            // Uncomment the line below to add a layout grid to the page
            //page.Elements.Add( new LayoutGrid() );

            // Add the page elements to the page
            AddAustraliaPost(page.Elements, 0, 0);
            AddDeutschePostIdentcode(page.Elements, 0, 120);
            AddDeutschePostLeitcode(page.Elements, 0, 240);
            AddKix(page.Elements, 0, 360);

            AddIntelligentMailBarcode(page.Elements, 279, 0);
            AddRm4sccCode(page.Elements, 279, 120);
            AddPostnet(page.Elements, 279, 240);
            AddSingaporePost(page.Elements, 279, 360);

            document.Pages.Add(page);
        }
```

```
        private void AddIsbnIssnIsmnPage(Document document, Outline parentOutline)
        {
            // Increments a page counter
            pageCounter++;

            // Adds to outlines to the page
            parentOutline.ChildOutlines.Add("Page " + pageCounter + " - ISBN/ISMN/ISSN Barcodes", new ZoomDestination(pageCounter, PageZoom.FitPage));

            // Create a page to add to the document
            ceTe.DynamicPDF.Page page = new ceTe.DynamicPDF.Page(PageSize.Letter, PageOrientation.Landscape, 54);

            // Uncomment the line below to add a layout grid to the page
            //page.Elements.Add( new LayoutGrid() );

            // Add the page elements to the page
            AddIsbn(page.Elements, 0, 0);
            AddIsbnSup2(page.Elements, 240, 0);
            AddIsbnSup5(page.Elements, 480, 0);
            AddIsmn(page.Elements, 0, 150);
            AddIsmnSup2(page.Elements, 240, 150);
            AddIsmnSup5(page.Elements, 480, 150);
            AddIssn(page.Elements, 0, 300);
            AddIssnSup2(page.Elements, 240, 300);
            AddIssnSup5(page.Elements, 480, 300);

            // Add the page to the document
            document.Pages.Add(page);
        }
        private void AddChartsPage(Document document, Outline parentOutline)
        {
            // Increments a page counter
            pageCounter++;

            // Adds  outlines to the page
            parentOutline.ChildOutlines.Add("Page " + pageCounter + " - Charts", new ZoomDestination(pageCounter, PageZoom.FitPage));

            // Create a page
            ceTe.DynamicPDF.Page page = new ceTe.DynamicPDF.Page(PageSize.Letter, PageOrientation.Landscape);
            AddAreaChart(page.Elements, 0, 25);
            AddLineChart(page.Elements, 375, 25);
            AddPieChart(page.Elements, 0, 260);
            AddBarChart(page.Elements, 375, 260);
            document.Pages.Add(page);
        }
        private void AddAreaChart(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "Stacked Area Chart", x, y, 325, 225);

```

```
            // Create a chart
            Chart chart = new Chart(x + 20, y + 25, 300, 200);

            // Create a plot area
            PlotArea plotArea = chart.PrimaryPlotArea;

            // Create header title and add it to the chart
            Title title1 = new Title("Website Visitors");
            chart.HeaderTitles.Add(title1);

            // Create indexed stacked area series elements and add values to it
            IndexedStackedAreaSeriesElement seriesElement1 = new IndexedStackedAreaSeriesElement("Website A");
            seriesElement1.Values.Add(new float[] { 5, 7, 9, 6 });
            IndexedStackedAreaSeriesElement seriesElement2 = new IndexedStackedAreaSeriesElement("Website B");
            seriesElement2.Values.Add(new float[] { 4, 2, 5, 8 });
            IndexedStackedAreaSeriesElement seriesElement3 = new IndexedStackedAreaSeriesElement("Website C");
            seriesElement3.Values.Add(new float[] { 2, 4, 6, 9 });

            // Create autogradient and add it to the series
            AutoGradient autogradient1 = new AutoGradient(90f, CmykColor.Red, CmykColor.IndianRed);
            seriesElement1.Color = autogradient1;
            AutoGradient autogradient2 = new AutoGradient(90f, CmykColor.Green, CmykColor.YellowGreen);
            seriesElement2.Color = autogradient2;
            AutoGradient autogradient3 = new AutoGradient(120f, CmykColor.Blue, CmykColor.LightBlue);
            seriesElement3.Color = autogradient3;

            // Create a Indexed Stacked Area Series
            IndexedStackedAreaSeries areaSeries = new IndexedStackedAreaSeries();

            // Add indexed stacked area series elements to the Indexed Stacked Area Series
            areaSeries.Add(seriesElement1);
            areaSeries.Add(seriesElement2);
            areaSeries.Add(seriesElement3);

            // Add series to the plot area
            plotArea.Series.Add(areaSeries);

            // Create a title and add it to the YAxis
            Title lTitle = new Title("Visitors (in millions)");
            areaSeries.YAxis.Titles.Add(lTitle);

            //Adding AxisLabels to the XAxis
            areaSeries.XAxis.Labels.Add(new IndexedXAxisLabel("Q1", 0));
            areaSeries.XAxis.Labels.Add(new IndexedXAxisLabel("Q2", 1));
            areaSeries.XAxis.Labels.Add(new IndexedXAxisLabel("Q3", 2));
            areaSeries.XAxis.Labels.Add(new IndexedXAxisLabel("Q4", 3));
            elements.Add(chart);
        }
        private void AddLineChart(Group elements, float x, float y)
        {
```

```
            AddCaptionAndRectangle(elements, "Line Chart", x, y, 325, 225);

            // Create a chart
            Chart chart = new Chart(x + 20, y + 25, 300, 200);

            // Create a plot area
            PlotArea plotArea = chart.PrimaryPlotArea;

            // Create header title and add it to the chart
            Title title1 = new Title("Website Visitors");
            chart.HeaderTitles.Add(title1);

            // Create a indexed line series and add values to it
            IndexedLineSeries lineSeries1 = new IndexedLineSeries("Website A");
            lineSeries1.Values.Add(new float[] { 5, 7, 9, 6 });
            IndexedLineSeries lineSeries2 = new IndexedLineSeries("Website B");
            lineSeries2.Values.Add(new float[] { 4, 2, 5, 8 });
            IndexedLineSeries lineSeries3 = new IndexedLineSeries("Website C");
            lineSeries3.Values.Add(new float[] { 2, 4, 6, 9 });

            // Add indexed line series to the plot area
            plotArea.Series.Add(lineSeries1);
            plotArea.Series.Add(lineSeries2);
            plotArea.Series.Add(lineSeries3);

            // Create a title and add it to the yaxis
            Title lTitle = new Title("Visitors (in millions)");
            lineSeries1.YAxis.Titles.Add(lTitle);

            //Adding AxisLabels to the XAxis
            lineSeries1.XAxis.Labels.Add(new IndexedXAxisLabel("Q1", 0));
            lineSeries1.XAxis.Labels.Add(new IndexedXAxisLabel("Q2", 1));
            lineSeries1.XAxis.Labels.Add(new IndexedXAxisLabel("Q3", 2));
            lineSeries1.XAxis.Labels.Add(new IndexedXAxisLabel("Q4", 3));
            elements.Add(chart);

        }
        private void AddPieChart(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "Pie Chart", x, y, 325, 225);

            // Create a chart
            Chart chart = new Chart(x + 20, y + 25, 300, 200);

            // Add a plot area to the chart
            PlotArea plotArea = chart.PlotAreas.Add(50, 50, 300, 300);

            // Create the Header title and add it to the chart
            Title tTitle = new Title("Website Visitors (in millions)");
            chart.HeaderTitles.Add(tTitle);
```

```
            // Create autogradient and add it to the series
            AutoGradient autogradient1 = new AutoGradient(90f, CmykColor.Red, CmykColor.IndianRed);
            AutoGradient autogradient2 = new AutoGradient(90f, CmykColor.Green, CmykColor.YellowGreen);
            AutoGradient autogradient3 = new AutoGradient(120f, CmykColor.Blue, CmykColor.LightBlue);

            // Create a scalar datalabel
            ScalarDataLabel da = new ScalarDataLabel(true, false, false);

            // Create a pie series
            PieSeries pieSeries = new PieSeries();

            // Set scalar datalabel to the pie series
            pieSeries.DataLabel = da;

            // Add series to the plot area
            plotArea.Series.Add(pieSeries);

            //Add pie series elements to the pie series
            pieSeries.Elements.Add(27, "Website A");
            pieSeries.Elements.Add(19, "Website B");
            pieSeries.Elements.Add(21, "Website C");
            pieSeries.Elements[0].Color = autogradient1;
            pieSeries.Elements[1].Color = autogradient2;
            pieSeries.Elements[2].Color = autogradient3;
            elements.Add(chart);
        }
        private void AddBarChart(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "Stacked Bar Chart", x, y, 325, 225);

            // Create a chart
            Chart chart = new Chart(x + 20, y + 25, 300, 200);

            // Create a plot area
            PlotArea plotArea = chart.PrimaryPlotArea;

            // Create header title and add it to the chart
            Title title1 = new Title("Website Visitors");
            chart.HeaderTitles.Add(title1);

            // Create a indexed stacked bar series elements and add values to it
            IndexedStackedBarSeriesElement seriesElement1 = new IndexedStackedBarSeriesElement("Website A");
            seriesElement1.Values.Add(new float[] { 5, 7, 9, 6 });
            IndexedStackedBarSeriesElement seriesElement2 = new IndexedStackedBarSeriesElement("Website B");
            seriesElement2.Values.Add(new float[] { 4, 2, 5, 8 });
            IndexedStackedBarSeriesElement seriesElement3 = new IndexedStackedBarSeriesElement("Website C");
            seriesElement3.Values.Add(new float[] { 2, 4, 6, 9 });

            // Create a Indexed Stacked Bar Series
```

```
            IndexedStackedBarSeries barSeries = new IndexedStackedBarSeries();

            // Create autogradient and add it to the series
            AutoGradient autogradient1 = new AutoGradient(90f, CmykColor.Red, CmykColor.IndianRed);
            AutoGradient autogradient2 = new AutoGradient(90f, CmykColor.Green, CmykColor.YellowGreen);
            AutoGradient autogradient3 = new AutoGradient(90f, CmykColor.Blue, CmykColor.LightBlue);
            seriesElement1.Color = autogradient1;
            seriesElement2.Color = autogradient2;
            seriesElement3.Color = autogradient3;

            // Add indexed stacked bar series elements to the Indexed Stacked Bar Series
            barSeries.Add(seriesElement1);
            barSeries.Add(seriesElement2);
            barSeries.Add(seriesElement3);

            //Add series to the plot area
            plotArea.Series.Add(barSeries);

            // Create a title and add it to the xaxis
            Title lTitle = new Title("Visitors (in millions)");
            barSeries.XAxis.Titles.Add(lTitle);

            //Adding AxisLabels to the yaxis
            barSeries.YAxis.Labels.Add(new IndexedYAxisLabel("Q1", 0));
            barSeries.YAxis.Labels.Add(new IndexedYAxisLabel("Q2", 1));
            barSeries.YAxis.Labels.Add(new IndexedYAxisLabel("Q3", 2));
            barSeries.YAxis.Labels.Add(new IndexedYAxisLabel("Q4", 3));
            barSeries.XAxis.Labels.Angle = -65;
            elements.Add(chart);

        }


        private void AddCodabar(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "Codabar Barcode", x, y, 225, 80);
            BarCode barCode = new Codabar("A1234B", x, y + 15, 48);
            barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
            barCode.Y += (65 - barCode.Height) / 2;
            elements.Add(barCode);
        }

        private void AddCode128(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "Code 128 Barcode", x, y, 225, 80);
            BarCode barCode = new Code128("Code 128 Barcode.", x, y + 15, 48, 0.75f);
            barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
            barCode.Y += (65 - barCode.Height) / 2;
            elements.Add(barCode);
        }
```

```
        private void AddCode25(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "Code 2 of 5 Barcode", x, y, 225, 80);
            BarCode barCode = new Code25("1234567890", x, y + 15, 48);
            barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
            barCode.Y += (65 - barCode.Height) / 2;
            elements.Add(barCode);
        }

        private void AddCode39(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "Code 3 of 9 Barcode", x, y, 225, 80);
            BarCode barCode = new Code39("CODE 39", x, y + 15, 48);
            barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
            barCode.Y += (65 - barCode.Height) / 2;
            elements.Add(barCode);
        }

        private void AddInterleaved25(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "Interleaved 2 of 5 Barcode", x, y, 225, 80);
            BarCode barCode = new Interleaved25("1234567890", x, y + 15, 48);
            barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
            barCode.Y += (65 - barCode.Height) / 2;
            elements.Add(barCode);
        }

        private void AddIntelligentMailBarcode(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "Intelligent Mail Barcode", x, y, 225, 80);
            BarCode barcode = new IntelligentMailBarCode("01234567094987654321012345678 91", x, y + 15, true);
            barcode.Height = 35;
            barcode.X += (225 - barcode.GetSymbolWidth()) / 2;
            barcode.Y += (65 - barcode.Height) / 2;
            elements.Add(barcode);
        }

        private void AddPostnet(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "Postnet Barcode", x, y, 225, 80);
            BarCode barCode = new Postnet("20815470412", x, y + 15);
            barCode.X += (225 - barCode.GetSymbolWidth()) / 2;
            barCode.Y += (65 - barCode.Height) / 2;
            elements.Add(barCode);
        }

        private void AddPdf417(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "PDF417 Barcode", x, y, 225, 100);
```

```
            byte[] by = ASCIIEncoding.ASCII.GetBytes("PDF417 Barcode");
            Pdf417 pdfc = new Pdf417(by, x, y + 15, 2, 2);
            pdfc.X += (225 - pdfc.GetSymbolWidth()) / 2;
            pdfc.Y += (85 - pdfc.GetSymbolHeight()) / 2;
            elements.Add(pdfc);
        }

        private void AddPdf417Macro(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "PDF417 Macro Barcode", x, y, 225, 100);
            byte[] values5 = null;
            try
            {
                FileStream fi = new FileStream(Server.MapPath(" ../Text/Textdata.txt"), FileMode.Open, FileAccess.Read);
                long avail = fi.Length;
                values5 = new byte[avail];
                fi.Read(values5, 0, (int)avail);
                fi.Close();
            }
            catch (Exception e)
            {
                Console.WriteLine("EXCEPTION  " + e.Message);
            }

            // Create a MacroPdf417
            MacroPdf417 macroPdf417 = new MacroPdf417(values5, x, y + 15, 10, 15, 0.9f);
            macroPdf417.X += (225 - macroPdf417.GetSymbolWidth()) / 2;
            macroPdf417.Y += (85 - macroPdf417.GetSymbolHeight()) / 2;

            // Add macroPdf417 to page
            elements.Add(macroPdf417);

            // Retrieve the overflow barcode and place it at the appropriate location on the page
            macroPdf417 = macroPdf417.GetOverflowMacroPdf417(x, y + 124);
            macroPdf417.X += (225 - macroPdf417.GetSymbolWidth()) / 2;
            macroPdf417.Y += (85 - macroPdf417.GetSymbolHeight()) / 2;
            AddCaptionAndRectangle(elements, "PDF417 Overflow Macro Barcode", x, y += 108, 225, 100);
            elements.Add(macroPdf417);
        }

        private void AddDataMatrixBarcode(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "Data Matrix Barcode", x, y, 225, 100);
            string text = "The photograph attached was taken by the crew on board the Columbia.";

            // Create a data matrix bar code
            DataMatrixBarcode dataMatrixBarcode = new DataMatrixBarcode(text, x, y+15, DataMatrixSymbolSize.R32xC32,
DataMatrixEncodingType.Auto);
            //dataMatrixBarcode.XDimension = 2f;
            dataMatrixBarcode.X += (225 - dataMatrixBarcode.GetSymbolWidth()) / 2;
```

```
            dataMatrixBarcode.Y += (85 - dataMatrixBarcode.GetSymbolHeight()) / 2;

            // Add data matrix barcode to the page
            elements.Add(dataMatrixBarcode);

            AddCaptionAndRectangle(elements, "Data Matrix Overflow Barcode", x, y += 108, 225, 100);

            // Retrieve the overflow barcode and place it at the appropriate location on the page
            dataMatrixBarcode = dataMatrixBarcode.GetOverFlowDataMatrixBarcode(x, y + 15);
            //dataMatrixBarcode.XDimension = 1.75f;
            dataMatrixBarcode.X += (225 - dataMatrixBarcode.GetSymbolWidth()) / 2;
            dataMatrixBarcode.Y += (85 - dataMatrixBarcode.GetSymbolHeight()) / 2;
            elements.Add(dataMatrixBarcode);
        }
        private void AddQrCode(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "QR Code", x, y, 225, 100);
            byte[] by = ASCIIEncoding.ASCII.GetBytes("QR Code Barcode");
            QrCode qrCode = new QrCode(by, x, y + 15);
            qrCode.XDimension = 2.5f;
            qrCode.X += (225 - qrCode.GetSymbolWidth()) / 2;
            qrCode.Y += (85 - qrCode.GetSymbolHeight()) / 2;
            elements.Add(qrCode);
        }

        private void AddUPCEANPage(Document document, Outline parentOutline)
        {
            // Increments a page counter
            pageCounter++;

            // Adds to outlines to the page
            parentOutline.ChildOutlines.Add("Page " + pageCounter + " - UPC/EAN/JAN Barcodes", new ZoomDestination(pageCounter,
PageZoom.FitPage));

            // Create a page to add to the document
            ceTe.DynamicPDF.Page page = new ceTe.DynamicPDF.Page(PageSize.Letter, PageOrientation.Landscape, 54);

            // Uncomment the line below to add a layout grid to the page
            //page.Elements.Add( new LayoutGrid() );

            // Add the page elements to the page
            AddEAN13(page.Elements, 0, 0);
            AddEAN13Sup2(page.Elements, 240, 0);
            AddEAN13Sup5(page.Elements, 480, 0);
            AddUPCVersionA(page.Elements, 0, 135);
            AddUPCVersionASup2(page.Elements, 240, 135);
            AddUPCVersionASup5(page.Elements, 480, 135);
            AddEAN8(page.Elements, 0, 270);
            AddEAN8Sup2(page.Elements, 240, 270);
            AddEAN8Sup5(page.Elements, 480, 270);
```

```
            AddUPCVersionE(page.Elements, 0, 405);
            AddUPCVersionESup2(page.Elements, 240, 405);
            AddUPCVersionESup5(page.Elements, 480, 405);

            // Add the page to the document
            document.Pages.Add(page);
        }

        private void AddEAN13(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "EAN/JAN 13 Barcode", x, y, 204, 99);
            BarCode barCode = new Ean13("123456789012", x, y + 21);
            barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
            elements.Add(barCode);
        }

        private void AddEAN13Sup2(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "EAN/JAN 13 Barcode, 2 digit supplement", x, y, 204, 99);
            BarCode barCode = new Ean13Sup2("12345678901212", x, y + 21);
            barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
            elements.Add(barCode);
        }

        private void AddEAN13Sup5(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "EAN/JAN 13 Barcode, 5 digit supplement", x, y, 204, 99);
            BarCode barCode = new Ean13Sup5("12345678901212345", x, y + 21);
            barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
            elements.Add(barCode);
        }

        private void AddUPCVersionA(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "UPC Version A Barcode", x, y, 204, 99);
            BarCode barCode = new UpcVersionA("12345678901", x, y + 21);
            barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
            elements.Add(barCode);
        }

        private void AddUPCVersionASup2(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "UPC Version A Barcode, 2 digit supplement", x, y, 204, 99);
            BarCode barCode = new UpcVersionASup2("1234567890112", x, y + 21);
            barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
            elements.Add(barCode);
        }

        private void AddUPCVersionASup5(Group elements, float x, float y)
        {
```

```
            AddCaptionAndRectangle(elements, "UPC Version A Barcode, 5 digit supplement", x, y, 204, 99);
            BarCode barCode = new UpcVersionASup5("1234567890112345", x, y + 21);
            barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
            elements.Add(barCode);
        }

        private void AddEAN8(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "EAN/JAN 8 Barcode", x, y, 204, 99);
            BarCode barCode = new Ean8("12345670", x, y + 21);
            barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
            elements.Add(barCode);
        }

        private void AddEAN8Sup2(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "EAN/JAN 8 Barcode, 2 digit supplement", x, y, 204, 99);
            BarCode barCode = new Ean8Sup2("1234567012", x, y + 21);
            barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
            elements.Add(barCode);
        }

        private void AddEAN8Sup5(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "EAN/JAN 8 Barcode, 5 digit supplement", x, y, 204, 99);
            BarCode barCode = new Ean8Sup5("1234567012345", x, y + 21);
            barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
            elements.Add(barCode);
        }

        private void AddUPCVersionE(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "UPC Version E Barcode", x, y, 204, 99);
            BarCode barCode = new UpcVersionE("0123456", x, y + 21);
            barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
            elements.Add(barCode);
        }

        private void AddUPCVersionESup2(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "UPC Version E Barcode, 2 digit supplement", x, y, 204, 99);
            BarCode barCode = new UpcVersionESup2("012345612", x, y + 21);
            barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
            elements.Add(barCode);
        }

        private void AddUPCVersionESup5(Group elements, float x, float y)
        {
            AddCaptionAndRectangle(elements, "UPC Version E Barcode, 5 digit supplement", x, y, 204, 99);
            BarCode barCode = new UpcVersionESup5("012345612345", x, y + 21);
```

```
            barCode.X += (204 - barCode.GetSymbolWidth()) / 2;
            elements.Add(barCode);
        }

        private void AddPageElementPage(Document document, Outline parentOutline)
        {
            // Increments a page counter
            pageCounter++;

            // Adds to outlines to the page
            Outline pageOutline = parentOutline.ChildOutlines.Add("Page " + pageCounter);
            pageOutline.ChildOutlines.Add("Zoom - Fit Page", new ZoomDestination(pageCounter, PageZoom.FitPage));
            pageOutline.ChildOutlines.Add("Zoom - Fit Width", new ZoomDestination(pageCounter, PageZoom.FitWidth));

            // Create a page to add to the document
            ceTe.DynamicPDF.Page page = new ceTe.DynamicPDF.Page(PageSize.Letter, PageOrientation.Portrait, 54);

            // Uncomment the line below to add a layout grid to the page
            //page.Elements.Add( new LayoutGrid() );

            // Add the page elements to the page
            AddHeader(page.Elements);
            AddBookmark(page.Elements, 0, 41, pageOutline);
            AddCircle(page.Elements, 279, 41);
            AddHtmlArea(page.Elements, 0, 134);
            AddImage(page.Elements, 0, 227);
            AddLabel(page.Elements, 279, 227);
            AddLine(page.Elements, 0, 320);
            AddLink(page.Elements, 279, 320);
            AddPath(page.Elements, 0, 413);
            AddRectangle(page.Elements, 279, 413);
            AddTextArea(page.Elements, 0, 506);
            AddCustom(page.Elements, 0, 599);
            AddNote(page.Elements, 279, 599);

            // Add the page to the document
            document.Pages.Add(page);
        }

        private void AddHeader(Group pageElements)
        {
            // Adds a header to the pageElements
            string formattedHtml = "<html><head><title>Header</title><style>body{font: 10pt helvetica;text-
align:center}</style></head><body><i>Dynamic</i><b>PDF</b> Generator for .NET has over 50 built-in page " +
                "elements. Custom page elements can be built by creating a class that inherits from " +
                "<b>ceTe.DynamicPDF.PageElement</b>.</body></html>";
            HtmlArea html = new HtmlArea(formattedHtml, 0, 0, 520, 450);
            //FormattedTextArea formattedTextArea = new FormattedTextArea(formattedHtml, 0, 0, 504, 450, FontFamily.Helvetica,
12, false);
            //formattedTextArea.Style.Paragraph.Align = TextAlign.Center;
```

```
            pageElements.Add(html);
        }

        private void AddBookmark(Group pageElements, float x, float y, Outline parentOutline)
        {
            // Adds a bookmark to the page
            AddCaptionAndRectangle(pageElements, "Bookmark Page Element", x, y);
            pageElements.Add(new Bookmark("Bookmarked Text", x + 5, y + 20, parentOutline));
            pageElements.Add(new Label("This text is bookmarked.", x + 5, y + 20, 215, 10, Font.TimesRoman, 10));
        }

        private void AddCircle(Group pageElements, float x, float y)
        {
            // Adds a circlt to the pageElements
            AddCaptionAndRectangle(pageElements, "Circle Page Element", x, y);
            pageElements.Add(new Circle(x + 112.5f, y + 50f, 107.5f, 30f, RgbColor.Red, RgbColor.Blue, 2, LineStyle.DashLarge));
        }


        private void AddHtmlArea(Group pageElements, float x, float y)
        {
            // Adds a html area to the pageElements
            Uri uri = new Uri(Server.MapPath("../Html/AllPageElementSample.html"));
            HtmlArea htmlArea = new HtmlArea(uri, x + 5, y + 20, 215, 70);

            // Sets the indent property
            AddCaptionAndRectangle(pageElements, "HtmlArea Page Element", x, y);
            AddCaptionAndRectangle(pageElements, "HtmlArea Overflow Text", x + 279, y);
            pageElements.Add(htmlArea);

            // Set the html area object equal to the rest of the html area that did not fit
            // if all the html area did fit, GetOverflowHtmlArea will return null
            HtmlArea overflowHtmlArea = htmlArea.GetOverflowHtmlArea(x + 284, y + 20, 70);
            pageElements.Add(overflowHtmlArea);

        }

        private void AddImage(Group pageElements, float x, float y)
        {
            // Adds an image to the pageElements
            AddCaptionAndRectangle(pageElements, "Image Page Element", x, y);
            Image image = new Image(Server.MapPath("../Images/DPDFLogo.png"), x + 112.5f, y + 50f, 0.24f);

            // Image is sized and centered in the rectangle
            image.SetBounds(215, 60);
            image.VAlign = VAlign.Center;
            image.Align = Align.Center;
            pageElements.Add(image);
        }

```

```
        private void AddLabel(Group pageElements, float x, float y)
        {
            // Adds a label to the pageElements
            AddCaptionAndRectangle(pageElements, "Label & PageNumberingLabel Page Elements", x, y);
            string labelText = "Labels can be rotated";
            string numberText = "PageNumberingLabels contain page numbering: %%CP%% of %%TP%% pages.";
            Label label = new Label(labelText, x + 5, y + 12, 220, 80, Font.TimesRoman, 12, TextAlign.Center);
            label.Angle = 8;
            PageNumberingLabel pageNumLabel = new PageNumberingLabel(numberText, x + 5, y + 55, 220, 80, Font.TimesRoman, 12, TextAlign.Center);
            pageElements.Add(pageNumLabel);
            pageElements.Add(label);
        }

        private void AddLine(Group pageElements, float x, float y)
        {
            // Adds a line to the pageElements
            AddCaptionAndRectangle(pageElements, "Line Page Element", x, y);
            pageElements.Add(new Line(x + 5, y + 20, x + 220, y + 80, 3, RgbColor.Green));
            pageElements.Add(new Line(x + 220, y + 20, x + 5, y + 80, 3, RgbColor.Green));
        }

        private void AddLink(Group pageElements, float x, float y)
        {
            // Adds a link to the pageElements
            Font font = Font.TimesRoman;
            string text = "This is a link to DynamicPDF.com.";
            AddCaptionAndRectangle(pageElements, "Link Page Element", x, y);
            Label label = new Label(text, x + 5, y + 20, 215, 80, font, 12, RgbColor.Blue);
            label.Underline = true;
            Link link = new Link(x + 5, y + 20, font.GetTextWidth(text, 12),
                12 - font.GetDescender(12), new UrlAction("http://www.dynamicpdf.com"));
            pageElements.Add(label);
            pageElements.Add(link);
        }

        private void AddPath(Group pageElements, float x, float y)
        {
            // Adds a path to the pageElements
            ceTe.DynamicPDF.PageElements.Path path = new ceTe.DynamicPDF.PageElements.Path(x + 5, y + 20, RgbColor.Blue, RgbColor.Red, 2, LineStyle.Solid, true);
            path.SubPaths.Add(new LineSubPath(x + 215, y + 40));
            path.SubPaths.Add(new CurveToSubPath(x + 108, y + 80, x + 160, y + 80));
            path.SubPaths.Add(new CurveSubPath(x + 5, y + 40, x + 65, y + 80, x + 5, y + 60));
            AddCaptionAndRectangle(pageElements, "Path Page Element", x, y);
            pageElements.Add(path);
        }

        private void AddRectangle(Group pageElements, float x, float y)
        {
```

```
            // Adds a rectangle to the pageElements
            AddCaptionAndRectangle(pageElements, "Rectangle Page Element", x, y);
            pageElements.Add(new Rectangle(x + 5, y + 20, 215, 60, RgbColor.Green, RgbColor.Blue, 2));
        }

        private void AddTextArea(Group pageElements, float x, float y)
        {
            // Adds a text area to the pageElements
            string text = "The TextArea page element is great for efficient processign of text that does " +
                "not require advanced formatting. It offers basic control over leading and paragraph spacing and " +
                "include text continuation functionality. If all the text does not fit within the bounds of " +
                "the TextArea the HasOverflowText method will return a true value. The GetOverflowTextArea " +
                "method can then be called to create a new TextArea with the remaining text. This " +
                "functionality makes it simple to span large amounts of text over multiple TextAreas and pages. Text can also be rotated.";
            AddCaptionAndRectangle(pageElements, "TextArea Page Element", x, y);
            AddCaptionAndRectangle(pageElements, "TextArea Overflow Text", x + 279, y);
            TextArea textArea = new TextArea(text, x + 5, y + 20, 215, 60, Font.TimesRoman, 10);

            // Create an overflow text area for the overflow text
            TextArea overflowTextArea = textArea.GetOverflowTextArea(x + 284, y + 20);
            pageElements.Add(textArea);
            pageElements.Add(overflowTextArea);
        }

        private void AddCustom(Group pageElements, float x, float y)
        {
            // Adds a custom page element to the page
            AddCaptionAndRectangle(pageElements, "Custom Page Element", x, y);
            pageElements.Add(new StopSign(80, y + 20, 60, 60));
        }

        private void AddNote(Group pageElements, float x, float y)
        {
            // Add a note element to the page
            AddCaptionAndRectangle(pageElements, "Note Page Element", x, y);
            pageElements.Add(new Note("This is a note", x + 100, y + 35, 50, 50, NoteType.Note));
        }

        private void AddTableAndOther(Document document, Outline parentOutline)
        {
            // Increments a page counter
            pageCounter++;

            // Adds to outlines to the page
            parentOutline.ChildOutlines.Add("Page " + pageCounter + " - Table and Additional Page Elements", new ZoomDestination(pageCounter, PageZoom.FitPage));

            // Create a page to add to the document
            ceTe.DynamicPDF.Page page = new ceTe.DynamicPDF.Page(PageSize.Letter, PageOrientation.Portrait, 54);
```

```
            // Uncomment the line below to add a layout grid to the page
            //page.Elements.Add( new LayoutGrid() );

            // Add the Table and other elements to the page
            AddTable(page.Elements, 0, 0);
            AddListElements(page, 0, 225);
            AddAdditional(page.Elements, 0, 500);

            // Add the page to the document
            document.Pages.Add(page);
        }

        private void AddTable(Group pageElements, float x, float y)
        {
            Label captionLabel1 = new Label("Table Page Element", x, y, 225, 10, Font.HelveticaBold, 10);
            pageElements.Add(captionLabel1);
            Label captionLabel2 = new Label("Table Row Overflow", x + 279, y, 225, 10, Font.HelveticaBold, 10);
            pageElements.Add(captionLabel2);
            Table2 table = new Table2(x, y + 15, 225, 190, Font.TimesRoman, 10);
            table.RepeatColumnHeaderCount = 1;

            // Add columns to the table
            table.Columns.Add(95);
            table.Columns.Add(65);
            table.Columns.Add(65);

            // Add the Header Row to the table
            Row2 row1 = table.Rows.Add(40, Font.HelveticaBold, 12, RgbColor.Black, RgbColor.LightBlue);
            row1.CellDefault.Align = TextAlign.Center;
            row1.CellDefault.VAlign = VAlign.Center;
            row1.Cells.Add("Header 1");
            row1.Cells.Add("Header 2");
            row1.Cells.Add("Header 3");
            row1.Cells.Add("Header 4");
            row1.Cells.Add("Header 5");
            row1.Cells.Add("Header 6");
            row1.Cells.Add("Header 7");
            row1.Cells.Add("Header 8");
            row1.Cells.Add("Header 9");
            row1.Cells.Add("Header 10");
            row1.Cells.Add("Header 11");
            row1.Cells.Add("Header 12");
            row1.Cells.Add("Header 13");
            row1.Cells.Add("Header 14");
            row1.Cells.Add("Header 15");
            row1.Cells.Add("Header 16");
            int cell = 0;

            // Add the rest of the rows to the table
```

```
            AddRow(table, ref cell, 1);
            AddRow(table, ref cell, 2);
            AddRow(table, ref cell, 3);
            AddRow(table, ref cell, 4);
            AddRow(table, ref cell, 5);
            AddRow(table, ref cell, 6);
            AddRow(table, ref cell, 7);
            AddRow(table, ref cell, 8);
            AddRow(table, ref cell, 9);
            AddRow(table, ref cell, 10);
            AddRow(table, ref cell, 11);
            AddRow(table, ref cell, 12);
            AddRow(table, ref cell, 13);
            AddRow(table, ref cell, 14);
            AddRow(table, ref cell, 15);
            AddRow(table, ref cell, 16);

            // Create an overflow table for the overflow columns
            Table2 overflowTable = table.GetOverflowRows(x + 279, y + 15);
            pageElements.Add(table);
            pageElements.Add(overflowTable);
        }

        private void AddRow(Table2 table, ref int cellNum, int rowNum)
        {
            Row2 row = table.Rows.Add();
            Cell2 cell = row.Cells.Add("RowHeader " + rowNum, Font.HelveticaBold, 12,
                RgbColor.Black, RgbColor.LightPink, 1);
            cell.Align = TextAlign.Center;
            cell.VAlign = VAlign.Center;
            row.Cells.Add("Item " + cellNum++);
            row.Cells.Add("Item " + cellNum++);
        }

        private void AddAdditional(Group pageElements, float x, float y)
        {
            // Adds text explaining the additional page elements
            string text = "An <b><u>AreaGroup</u></b> represents a group of page elements with an area.<p/>" +
                "A <b><u>BackgroundImage</u></b> page element is used to add an image to the background of a page. <p/>" +
                "The <b><u>Group</u></b> page element is used for grouping page elements together.<p/>" +
                "The <b><u>LayoutGrid</u></b> draws a coordinate grid to the page.<p/>" +
                "A <b><u>TransformationGroup</u></b> can rotate and scale a group of page elements.<p/>" +
                "A <b><u>TransparencyGroup</u></b> can be used to add transparency to page elements.<p/>" +
                "An <b><u>Anchor Group </u></b> can have Page Elements within it. It can be anchored to the edge or margins of the page.<p/>";
            FormattedTextArea formattedTextArea = new FormattedTextArea(text, x + 5, y + 20, 494, 180, FontFamily.Times, 12, false);
            formattedTextArea.Style.Paragraph.SpacingAfter = 5;
            AddCaptionAndRectangle(pageElements, "Additional Page Elements", x, y, 504, 180);
            pageElements.Add(formattedTextArea);
```

```
        }

        private void AddCaptionAndRectangle(Group pageElements, string caption, float x, float y, float width, float height)
        {
            // Adds a rectangle and caption to the pageElements
            Rectangle rectangle = new Rectangle(x, y + 15, width, height - 15);
            Label captionLabel = new Label(caption, x, y, 225, 10, Font.HelveticaBold, 10);
            pageElements.Add(rectangle);
            pageElements.Add(captionLabel);
        }

        private void AddCaptionAndRectangle(Group pageElements, string caption, float x, float y)
        {
            AddCaptionAndRectangle(pageElements, caption, x, y, 225, 85);
        }

        private void AddFormElementPage(Document document, Outline parentOutline)
        {
            pageCounter++;
            parentOutline.ChildOutlines.Add("Page " + pageCounter + " - Form Page Elements", new ZoomDestination(pageCounter, PageZoom.FitPage));

            // Create a page to add to the document
            ceTe.DynamicPDF.Page page = new ceTe.DynamicPDF.Page(PageSize.Letter, PageOrientation.Portrait, 54);
            AddTextField(page.Elements, 0, 0);
            AddComboField(page.Elements, 0, 105);
            AddListField(page.Elements, 0, 210);
            AddRadioField(page.Elements, 0, 315);
            AddCheckBoxField(page.Elements, 0, 420);
            AddButtonField(page.Elements, 0, 525);

            // Add the page to the document
            document.Pages.Add(page);
        }

        private void AddListElements(Page page, float x, float y)
        {
            OrderedList orderedList = new OrderedList(x + 5, y + 20, 400, 90, Font.Helvetica, 10);
            orderedList.Items.Add("Fruits");
            orderedList.Items.Add("Vegetables");

            OrderedSubList orderedSubList = orderedList.Items[0].SubLists.AddOrderedSubList(NumberingStyle.RomanUpperCase);
            orderedSubList.Items.Add("Citrus");
            orderedSubList.Items.Add("Non-Citrus");

            AddCaptionAndRectangle(page.Elements, "Ordered List Page Element", x, y, 225, 110);
            OrderedSubList orderedSubList2 = orderedList.Items[1].SubLists.AddOrderedSubList(NumberingStyle.RomanUpperCase);
            orderedSubList2.Items.Add("Potato");
            orderedSubList2.Items.Add("Beans");

```

```
            OrderedSubList subOrderedSubList =

            subOrderedSubList.Items.Add("Lime");
            subOrderedSubList.Items.Add("Orange");

            OrderedSubList subOrderedSubList2 =
orderedSubList.Items[1].SubLists.AddOrderedSubList(NumberingStyle.RomanLowerCase);
            subOrderedSubList2.Items.Add("Mango");
            subOrderedSubList2.Items.Add("Banana");

            OrderedSubList subOrderedSubList3 =
orderedSubList2.Items[0].SubLists.AddOrderedSubList(NumberingStyle.RomanLowerCase);
            subOrderedSubList3.Items.Add("Sweet Potato");

            OrderedSubList subOrderedSubList4 =
orderedSubList2.Items[1].SubLists.AddOrderedSubList(NumberingStyle.RomanLowerCase);
            subOrderedSubList4.Items.Add("String Bean");
            subOrderedSubList4.Items.Add("Lima Bean");
            subOrderedSubList4.Items.Add("Kidney Bean");

            page.Elements.Add(orderedList);
            x += 279;

            orderedList = orderedList.GetOverFlowList(x + 5, y + 20);
            AddCaptionAndRectangle(page.Elements, "Ordered List Page Element Overflow", x, y, 225, 110);
            page.Elements.Add(orderedList);

            x = 0;
            y += 138;

            // Create an unordered list
            UnorderedList unorderedList = new UnorderedList(x + 5, y + 20, 400, 90, Font.Helvetica, 10);
            unorderedList.Items.Add("Fruits");
            unorderedList.Items.Add("Vegetables");

            UnorderedSubList unorderedSubList = unorderedList.Items[0].SubLists.AddUnorderedSubList();
            unorderedSubList.Items.Add(" Citrus");
            unorderedSubList.Items.Add(" Non-Citrus");

            AddCaptionAndRectangle(page.Elements, "Unordered List Page Element", x, y, 225, 110);

            UnorderedSubList unorderedSubList2 = unorderedList.Items[1].SubLists.AddUnorderedSubList();
            unorderedSubList2.Items.Add("Potato");
            unorderedSubList2.Items.Add("Beans");

            UnorderedSubList subUnorderedSubList = unorderedSubList.Items[0].SubLists.AddUnorderedSubList();
            subUnorderedSubList.Items.Add("Lime");
            subUnorderedSubList.Items.Add("Orange");

            UnorderedSubList subUnorderedSubList2 = unorderedSubList.Items[1].SubLists.AddUnorderedSubList();
```

```
            subUnorderedSubList2.Items.Add("Mango");
            subUnorderedSubList2.Items.Add("Banana");

            UnorderedSubList subUnorderedSubList3 = unorderedSubList2.Items[0].SubLists.AddUnorderedSubList();
            subUnorderedSubList3.Items.Add("Sweet Potato");

            UnorderedSubList subUnorderedSubList4 = unorderedSubList2.Items[1].SubLists.AddUnorderedSubList();
            subUnorderedSubList4.Items.Add("String Bean");
            subUnorderedSubList4.Items.Add("Lima Bean");
            subUnorderedSubList4.Items.Add("Kidney Bean");

            x += 279;
            page.Elements.Add(unorderedList);
            unorderedList = unorderedList.GetOverFlowList(x + 5, y + 20);
            AddCaptionAndRectangle(page.Elements, "Unordered List Page Element Overflow", x, y, 225, 110);
            page.Elements.Add(unorderedList);
        }

        private void AddTextField(Group pageElements, float x, float y)
        {
            TextField txt = new TextField("txtfname", x + 20, y + 40, 120, 20);
            txt.DefaultValue = "This is a Scrollable TextField";
            txt.BorderColor = RgbColor.Black;
            txt.BackgroundColor = RgbColor.AliceBlue;
            txt.ToolTip = "Scrollable";
            pageElements.Add(txt);

            TextField txt1 = new TextField("txtf1name", x + 175, y + 40, 120, 20);
            txt1.DefaultValue = "TextField";
            txt1.Password = true;
            txt1.MaxLength = 9;
            txt1.BorderColor = RgbColor.Black;
            txt1.BackgroundColor = RgbColor.AliceBlue;
            txt1.ToolTip = "Password with Maximum Length";
            pageElements.Add(txt1);

            TextField txt2 = new TextField("txtf2name", x + 330, y + 30, 150, 40);
            txt2.DefaultValue = "This is a TextField which goes to the next line if the text exceeds width";
            txt2.MultiLine = true;
            txt2.BorderColor = RgbColor.Black;
            txt2.BackgroundColor = RgbColor.AliceBlue;
            txt2.ToolTip = "Multiline";
            pageElements.Add(txt2);

            AddCaptionAndRectangle(pageElements, "TextField Form Page Element", x, y, 504, 85);
        }

        private void AddComboField(Group pageElements, float x, float y)
        {
            ComboBox cb = new ComboBox("cmbname", x + 51, y + 40, 150, 20);
```

```
            cb.BorderColor = RgbColor.Black;
            cb.BackgroundColor = RgbColor.AliceBlue;
            cb.Font = Font.Helvetica;
            cb.FontSize = 12;
            cb.Items.Add("Item 1");
            cb.Items.Add("Item 2");
            cb.Items.Add("Item 3");
            cb.Items.Add("Item 4");
            cb.Items.Add("Editable");
            cb.Items["Editable"].Selected = true;
            cb.Editable = true;
            cb.ToolTip = "Editable Combo Box";
            pageElements.Add(cb);

            ComboBox cb1 = new ComboBox("cmb1name", x + 303, y + 40, 150, 20);
            cb1.BorderColor = RgbColor.Black;
            cb1.BackgroundColor = RgbColor.AliceBlue;
            cb1.Font = Font.Helvetica;
            cb1.FontSize = 12;
            cb1.Items.Add("Item 1");
            cb1.Items.Add("Item 2");
            cb1.Items.Add("Item 3");
            cb1.Items.Add("Item 4");
            cb1.Items.Add("Non-Editable");
            cb1.Items["Non-Editable"].Selected = true;
            cb1.Editable = false;
            cb1.ToolTip = "Non-Editable Combo Box";
            pageElements.Add(cb1);

            AddCaptionAndRectangle(pageElements, "ComboBox Form Page Element", x, y, 504, 85);
        }

        private void AddListField(Group pageElements, float x, float y)
        {
            ListBox lb = new ListBox("lbname", x + 51, y + 30, 150, 40);
            lb.BorderColor = RgbColor.Black;
            lb.BackgroundColor = RgbColor.AliceBlue;
            lb.Font = Font.Helvetica;
            lb.FontSize = 12;
            lb.Items.Add("Item 1");
            lb.Items.Add("Item 2");
            lb.Items.Add("Item 3");
            lb.Items.Add("Item 4");
            lb.Items.Add("Non-MultiSelect");
            lb.Items["Non-MultiSelect"].Selected = true;
            lb.Multiselect = false;
            lb.ToolTip = "Simple List Box";
            pageElements.Add(lb);

            ListBox lb1 = new ListBox("lb1name", x + 303, y + 30, 150, 40);
```

```
            lb1.BorderColor = RgbColor.Black;
            lb1.BackgroundColor = RgbColor.AliceBlue;
            lb1.Font = Font.Helvetica;
            lb1.FontSize = 12;
            lb1.Items.Add("Item 1");
            lb1.Items.Add("Item 2");
            lb1.Items.Add("Item 3");
            lb1.Items.Add("Item 4");
            lb1.Items["Item 1"].Selected = true;
            lb1.Items.Add("MultiSelect");
            lb1.Items["MultiSelect"].Selected = true;
            lb1.Multiselect = true;
            lb1.ToolTip = "MultiSelect List Box";
            pageElements.Add(lb1);

            AddCaptionAndRectangle(pageElements, "ListBox Form Page Element", x, y, 504, 85);
        }

        private void AddRadioField(Group pageElements, float x, float y)
        {
            RadioButton rb = new RadioButton("rbname", x + 40, y + 42, 12, 12);
            rb.DefaultChecked = true;
            rb.BackgroundColor = RgbColor.AliceBlue;
            rb.TextColor = RgbColor.Red;
            rb.ExportValue = "Radio1";
            Label lb = new Label("Monthly", x + 75, y + 43, 60, 15);
            pageElements.Add(rb);
            pageElements.Add(lb);

            RadioButton rb1 = new RadioButton("rbname", x + 195, y + 42, 12, 12);
            rb1.BackgroundColor = RgbColor.AliceBlue;
            rb1.TextColor = RgbColor.Green;
            rb1.ExportValue = "Radio2";
            Label lb1 = new Label("Quarterly", x + 235, y + 43, 60, 15);
            pageElements.Add(rb1);
            pageElements.Add(lb1);

            RadioButton rb2 = new RadioButton("rbname", x + 350, y + 42, 12, 12);
            rb2.BackgroundColor = RgbColor.AliceBlue;
            rb2.TextColor = RgbColor.Blue;
            rb2.ExportValue = "Radio3";
            Label lb2 = new Label("Yearly", x + 385, y + 43, 60, 15);
            pageElements.Add(rb2);
            pageElements.Add(lb2);

            AddCaptionAndRectangle(pageElements, "RadioButton Form Page Element", x, y, 504, 85);
        }

        private void AddCheckBoxField(Group pageElements, float x, float y)
        {
```

```
            CheckBox cb = new CheckBox("cbxname", x + 40, y + 42, 15, 15);
            cb.Symbol = ceTe.DynamicPDF.PageElements.Forms.Symbol.Check;
            cb.BackgroundColor = RgbColor.AliceBlue;
            Label lb = new Label("Onion", x + 75, y + 43, 60, 15);
            pageElements.Add(cb);
            pageElements.Add(lb);

            CheckBox cb1 = new CheckBox("cbxsecondname", x + 195, y + 42, 15, 15);
            cb1.Symbol = ceTe.DynamicPDF.PageElements.Forms.Symbol.Cross;
            cb1.DefaultChecked = true;
            cb1.BackgroundColor = RgbColor.AliceBlue;
            Label lb1 = new Label("Capsicum", x + 235, y + 43, 60, 15);
            pageElements.Add(cb1);
            pageElements.Add(lb1);

            CheckBox cb2 = new CheckBox("cbxthirdname", x + 350, y + 42, 15, 15);
            cb2.Symbol = ceTe.DynamicPDF.PageElements.Forms.Symbol.Star;
            cb2.BackgroundColor = RgbColor.AliceBlue;
            Label lb2 = new Label("Jalapenos", x + 385, y + 43, 80, 15);
            pageElements.Add(cb2);
            pageElements.Add(lb2);

            AddCaptionAndRectangle(pageElements, "CheckBox Form Page Element", x, y, 504, 85);
        }

        private void AddButtonField(Group pageElements, float x, float y)
        {
            Button bt = new Button("btonename", x + 40, y + 40, 100, 25);
            bt.Behavior = Behavior.Push;
            bt.BorderStyle = BorderStyle.Beveled;
            SubmitAction sb = new SubmitAction(Request.Path);
            bt.ToolTip = "Submit Button";
            bt.Label = "Submit";
            bt.Action = sb;
            pageElements.Add(bt);

            Button bt1 = new Button("bttwoname", x + 195, y + 40, 100, 25);
            bt1.Behavior = Behavior.Push;
            bt1.BorderStyle = BorderStyle.Beveled;
            ResetAction rb = new ResetAction();
            bt1.Action = rb;
            bt1.ToolTip = "Reset Button";
            bt1.Label = "Reset";
            pageElements.Add(bt1);

            Button bt2 = new Button("btthreename", x + 350, y + 40, 100, 25);
            bt2.Behavior = Behavior.Push;
            bt2.BorderStyle = BorderStyle.Beveled;
            bt2.Action = new ZoomDestination(pageCounter, PageZoom.FitWidth);
            bt2.Behavior = Behavior.CreatePush("Click", "Zoom");
```

```
            bt2.ToolTip = "Zoom to fit page width";
            bt2.Label = "Zoom";
            pageElements.Add(bt2);

            Button bt3 = new Button("btfourname", x + 101, y + 90, 100, 25);
            bt3.Behavior = Behavior.Push;
            bt3.BorderStyle = BorderStyle.Beveled;
            bt3.Action = new XYDestination(pageCounter, 100, 100);
            bt3.Behavior = Behavior.CreatePush("Click", "Go To XY Destination");
            bt3.ToolTip = "Go to XY position";
            bt3.Label = "XY";
            pageElements.Add(bt3);

            Button bt4 = new Button("btfivename", x + 302, y + 90, 100, 25);
            bt4.Behavior = Behavior.Push;
            bt4.BorderStyle = BorderStyle.Beveled;
            bt4.Action = new JavaScriptAction("app.alert('Hello');");
            bt4.ToolTip = "Javascript Button";
            bt4.Label = "OK";
            pageElements.Add(bt4);

            AddCaptionAndRectangle(pageElements, "Button Form Page Element", x, y, 504, 140);
        }

        private void AddWatermark(Document document, Outline parentOutline)
        {
            // Increments a page counter
            pageCounter++;

            // Adds to outlines to the page
            parentOutline.ChildOutlines.Add("Page " + pageCounter + " - Watermark text", new ZoomDestination(pageCounter,
PageZoom.FitPage));

            // Create a page to add to the document
            ceTe.DynamicPDF.Page page = new ceTe.DynamicPDF.Page(PageSize.Letter, PageOrientation.Portrait, 54);

            // Create Text watermark
            TextWatermark watermark = new TextWatermark("Draft");
            watermark.TextColor = null;
            watermark.TextOutlineColor = RgbColor.Black;

            string caption = "Text and Image Watermark Page Elements";
            Label captionLabel = new Label(caption, 0, 235, 225, 10, Font.HelveticaBold, 10);
            page.Elements.Add(captionLabel);

            // Create Label and add discription text
            string text = "Text or Image watermarks can be placed on any page";
            Label label = new Label(text, 5, 255, 500, 100);
            label.FontSize = 40;
            page.Elements.Add(label);
```

```
1626
1627            // Create a Rectangle
1628            Rectangle rect = new Rectangle(0,250,500,100);
1629            page.Elements.Add(rect);
1630
1631            // Add Text watermark to page
1632            page.Elements.Add(watermark);
1633
1634            // Set Pdf version to 1.6
1635            document.PdfVersion = PdfVersion.v1_6;
1636
1637            // Add page to document
1638            document.Pages.Add(page);
1639
1640        }
1641
1642        #endregion
1643
1644        #region Web Form Designer generated code
1645        override protected void OnInit(EventArgs e)
1646        {
1647            //
1648            // CODEGEN: This call is required by the ASP.NET Web Form Designer.
1649            //
1650            InitializeComponent();
1651            base.OnInit(e);
1652        }
1653
1654        /// <summary>
1655        /// Required method for Designer support - do not modify
1656        /// the contents of this method with the code editor.
1657        /// </summary>
1658        private void InitializeComponent()
1659        {
1660        }
1661        #endregion
1662    }
1663 }
```