FormFieldReader.aspx

```
1 <%@ Page language="c#" Inherits="Web_CSharp.MergerExamples.FormFieldReader" CodeFile="FormFieldReader.aspx.cs" %>
2
```

FormFieldReader.aspx.cs

```
using System;
using ceTe.DynamicPDF;
using ceTe.DynamicPDF.Merger;
using ceTe.DynamicPDF.Merger.Forms;
using ceTe.DynamicPDF.PageElements;

namespace Web_CSharp.MergerExamples
{
    public partial class FormFieldReader : System.Web.UI.Page
    {
        protected void Page_Load(object sender, System.EventArgs e)
        {
            // PDF to read form fields from
            PdfDocument pdfDocument = new PdfDocument( MapPath("../PDFs/fw9AcroForm_14_filled.pdf") );

            // Create Table to display the form fields and values
            Table2 table = new Table2( 0, 0, 512, 676, ceTe.DynamicPDF.Font.Helvetica, 12 );
            table.Border.Width = 1;
            table.Border.Color = RgbColor.Black;
            table.RepeatColumnHeaderCount = 1;
            table.RepeatRowHeaderCount = 1;
            BuildTable( table );
            CreateList( table, pdfDocument.Form.Fields );

            Document document = new Document();
            AddTableToPage( document, table );

            Table2 overflowRowTable = table.GetOverflowRows();
                while( overflowRowTable != null )
            {
                AddTableToPage( document, overflowRowTable );
                overflowRowTable = overflowRowTable.GetOverflowRows();
            }
            document.DrawToWeb( "FormFieldReader.pdf" );
        }

        private void BuildTable( Table2 table )
        {
            CreateColumns( table );
            CreateRowHeadings( table );
        }

        private void CreateList( Table2 table, ceTe.DynamicPDF.Merger.Forms.PdfFormFieldList fieldList )
        {
            for( int i = 0; i < fieldList.Count; i++ )
            {
                Row2 row = table.Rows.Add( 20 );
                row.Cells.Add( fieldList[i].FullName );
                row.Cells.Add( GetFieldType(fieldList[i]) );
                row.Cells.Add( fieldList[i].GetValue() );
```

```
                if( fieldList[i].HasChildFields == true )
                {
                    CreateList( table, fieldList[i].ChildFields );
                }
            }
        }

        private string GetFieldType(PdfFormField pdfFormField)
        {
            if(pdfFormField is PdfTextField)
            {
                return "Text Field";
            }
            if(pdfFormField is PdfButtonField)
            {
                return "Button Field";
            }
            if(pdfFormField is PdfChoiceField)
            {
                return "Choice Field";
            }
            if(pdfFormField is PdfSignatureField)
            {
                return "Signature Field";
            }
            return "Container Field";
        }

        private void AddTableToPage( Document document, Table2 table )
        {
            Page page = new Page( PageSize.Letter );
            if( table != null )
                page.Elements.Add( table );
            document.Pages.Add( page );
        }

        private void CreateRowHeadings( Table2 table )
        {
            Row2 row = table.Rows.Add( 40, Font.TimesBold, 12, RgbColor.Black, RgbColor.LightGrey );
            row.CellDefault.Align  = TextAlign.Center;
            row.CellDefault.VAlign  = VAlign.Top;
            row.Cells.Add( "FormField Name" );
            row.Cells.Add( "FormField Type" );
            row.Cells.Add( "FormField Value" );
        }

        private void CreateColumns( Table2 table )
        {
            table.Columns.Add( 150 );
            table.Columns.Add( 150 );
```

```
            table.Columns.Add( 212 );
        }

        #region Web Form Designer generated code
        override protected void OnInit(EventArgs e)
        {
            //
            // CODEGEN: This call is required by the ASP.NET Web Form Designer.
            //
            InitializeComponent();
            base.OnInit(e);
        }

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
        }
        #endregion
    }
}
```