JOINANDSHARE
Customer Software Solutions

Aspose.Total for .NET

Case Study

Use of Aspose.Total for the defined transformation of various files and formats into PDFs

Thomas Groß, Software Engineer

About join and share

Our company is located in Berlin, Germany. We develop customer-specific solutions based first and foremost on Microsoft products. Our certification for the Microsoft Silver Data Platform sets us apart as experts for database-oriented solutions in the field of business intelligence and enterprise applications.
Our team consists of Microsoft .NET experts, database developers for Microsoft SQL Server, Access and Office, Web developers and UX Designers.
Our customers are generally companies from the following sectors: transportation, insurance, housing, health care, education, and public administration.

Problem

The Insurance Ombudsman is one of our long-term customers. It helps individual policyholders to have their complaints settled out of the courts system in a cost-effective, efficient and impartial way. In the past, documents for the arbitration process had to be sent by mail or fax. The new goal was that requests for the arbitration can be made on a web portal.

For the request, the complainant is able to upload documents in a wide range of text and image file formats. The documents are first stored in a database and then combined in PDFs. In the file, all pages are marked (stamped) with the name and the file reference of the complaint. Finally, the requests are saved and printed as single PDF files.

A combination of open source and/or commercially available products is not the first choice for our development team because we don't want to run into problems regarding the fine control and settings of the PDF produced. The requirement is to use a time-effective and easy solution.

Solution

Unhesitatingly, we decided to evaluate Aspose.Total for the data processing of the various uploaded files.

The decision to use Aspose.Total was quickly made because we had very good results with the performance and flexibility of Aspose.Words in other projects. In our experience, the Aspose product family covers almost any feature one can have related to documents with a lot of additional functionality that would have otherwise taken a lot of time / cost to develop.

Some interesting aspects of the implementation:

Uncompressed image files are converted to JPEG with the component Aspose.Image.

```
    break;
case ".img":
case ".tif":
case ".tiff":
case ".bmp":
case ".png":
case ".gif":
    output2 = new FileStream(ConfigurationManager.AppSettings["TempImage"] + strExtension.ToLower(), FileMode.Create, FileAccess.Write);
    output2.Write(blob, 0, blob.Length);
    output2.Close();
    if (Aspose.Imaging.Image.CanLoad(ConfigurationManager.AppSettings["TempImage"] + strExtension.ToLower()))
    {
        Aspose.Imaging.Image myImage = Aspose.Imaging.Image.Load(ConfigurationManager.AppSettings["TempImage"] + strExtension.ToLower());
        myImage.Save(ConfigurationManager.AppSettings["TempImage"] + ".jpg");
        myImage.Dispose();
        JpegImageSeiteErstellen(ConfigurationManager.AppSettings["TempImage"] + ".jpg", pdfStreams[i]);
    }
    else
```

The requirement was that the uploaded image files are printed on A4 sized paper with defined resolution, picture quality and margins. Easily and simply done with Aspose.PDF:

```
// Add page to pages collection of PDF file
Page page = pdfdoc.Pages.Add();
page.PageInfo.Margin.Left = page.PageInfo.Margin.Right = page.PageInfo.Margin.Bottom = page.PageInfo.Margin.Top = 0;
page.AddImage(ConfigurationManager.AppSettings["TempImage"] + "2.jpg",
    new Aspose.Pdf.Rectangle(abstand_x * 72 / 25.4, abstand_y * 72 / 25.4, (abstand_x + bildbreite_mm) * 72 / 25.4, (abstand_y + bildhoehe_mm) * 72 / 25.4));
pdfdoc.Save(target);
```

Pictures are easily rotated for an optimized positioning:

```
Aspose.Imaging.Image myImage = Aspose.Imaging.Image.Load(strPfad);
if (myImage.Size.Width > myImage.Size.Height)
{
    myImage.RotateFlip(Aspose.Imaging.RotateFlipType.Rotate90FlipNone);
}
myImage.Save(ConfigurationManager.AppSettings["TempImage"] + "2.jpg", BildQualitaet);
int bildbreite_mm = System.Convert.ToInt32((25.4 / System.Convert.ToDouble(ConfigurationManager.AppSettings["BildAufloesung"].ToString()) * myImage.Size.Width)
```

The text files (Word, OpenOffice, RTF, TXT,…) are quickly and safely converted to PDFs with Aspose.Words:

```
case ".doc":
case ".docx":
case ".rtf":
case ".odt":
case ".txt":
    MemoryStream textdocStream = new MemoryStream();
    textdocStream.Write(blob, 0, blob.Length);
    Aspose.Words.Document doc = new Aspose.Words.Document(textdocStream);
    doc.Save(pdfStreams[i], Aspose.Words.SaveFormat.Pdf);
    break;
```

Aspose.Cells does the same job with Excel files with only four lines of code:

```
MemoryStream cellsStream = new MemoryStream();
cellsStream.Write(blob, 0, blob.Length);
Aspose.Cells.Workbook cells = new Aspose.Cells.Workbook(cellsStream);
cells.Save(pdfStreams[i], Aspose.Cells.SaveFormat.Pdf);
```

The stamps for the documents are generated with Aspose.PDF:

```
Aspose.Pdf.Document pdfdoc = new Aspose.Pdf.Document(source);
foreach (Page page in pdfdoc.Pages)
{
    TextStamp textStamp = new TextStamp(strDateiname);
    textStamp.XIndent = System.Convert.ToSingle(ConfigurationManager.AppSettings["Wasserzeichen_Dateiname_X"].ToString()) * 72 / 25.4;
    textStamp.YIndent = System.Convert.ToSingle(ConfigurationManager.AppSettings["Wasserzeichen_Dateiname_Y"].ToString()) * 72 / 25.4; ;
    textStamp.Rotate = Rotation.on90;
    textStamp.TextState.Font = FontRepository.FindFont(ConfigurationManager.AppSettings["Wasserzeichen_Dateiname_Font"]);
    textStamp.TextState.FontSize = System.Convert.ToSingle(ConfigurationManager.AppSettings["Wasserzeichen_Dateiname_FontSize"].ToString());
    textStamp.TextState.ForegroundColor = Aspose.Pdf.Color.Black;
    page.AddStamp(textStamp);
```

Finally, the PDFs of a single query are transformed to a single PDF file, saved on a file server and printed out. Quickly achieved with the pdfEditor and pdfViewer object of Aspose.PDF:

```
MemoryStream StreamDiesesAZ = new MemoryStream();
PdfFileEditor pdfEditor = new PdfFileEditor();
pdfEditor.Concatenate(pdfStreams, StreamDiesesAZ);
```

```
pgs.PaperSize = new System.Drawing.Printing.PaperSize("A4", 827, 1169);
pgs.Margins = new System.Drawing.Printing.Margins(0, 0, 0, 0);

//Print document using printer and page settings
viewer.PrintDocumentWithSettings(pgs, ps);
```

Experience

We have developped the application with the evaluation version of Aspose.Total. As expected, we are highly content with the results and absolutely convinced to use the product in future projects as well.

Next Steps

We are planning to optimize further processes of the customer with Aspose.Total. One task will be the processing of incoming e-mails with attachments.

Summary

Aspose.Total is a powerful suite to process nearly all kinds of documents in high speed and great quality. It effectively reduces the programming effort.

The Aspose.Total components provide a clean and well-documented object-model. We also found many helpful code examples on the Aspose website.

We would highly recommend the components to others.