



CASE STUDY

Aspose.Total for .Net Case Study



Mainpac Solutions Pty Ltd

Using Aspose.Total for .Net to generate a merged Work Order PDF document.

Angelo Vargheese, Development Manager, 05 July 2021

About Mainpac Solutions

Established in 1984, Mainpac is an Australian software vendor which is now part of the Vela Software Group, a wholly owned subsidiary of Constellation Software, Inc. (TSX:CSU), an international provider of software and services to multiple industries across public and private sectors.

Mainpac provides Enterprise Asset Management (EAM) solutions across Australia and within 23 international markets to asset-intensive industries including manufacturing, mining and resources, ports and terminals, utilities, government, and facilities management. Mainpac's EAM on-premise and on-demand software solutions are complemented by consulting, implementation, training, and support services.

Mainpac EAM allows planning and control of all the activities required to maintain business assets throughout each individual asset's lifecycle. These activities include, but are not limited to, the scheduling of planned maintenance, optimization of asset productivity, documentation of information in relation to breakdown and malfunctions, stock control, inventory, purchasing information and many more features.

Problem

In Mainpac EAM users can attach documents, diagrams, and pictures to a Work Order (Job). These documents provide important information to field technicians to help them complete their work safely and efficiently.

When a Work Order is issued to a field technician, using the application the field technician can print a Microsoft Reporting Service report with the Work Order details and then print the individual attached documents.

Several of our customers have requested that when a work order is issued to a field technician that they get sent an email with a single PDF document containing the details of the Work Order and all attached documents.

Depending on the organization and industry these documents can be in several different formats including PDF, Text, Word, Excel, Visio, AutoCAD DWG, PNG, and JPEG.

To complete this enhancement, we need a component that could generate PDF from multiple file formats and be able to merge these documents with the Work Order report.

After visiting the Aspose website we thought that Aspose.Total product suite could be a solution and decided to evaluate the product.

How important was it to solve these problems?

Customers have noted that not having all the relevant information for a work order in one document as critical, as it would impact the safety and the time to complete a work order.

How many users were involved, how many customers affected?

Multiple customers with up to 200 users each, generating between 200 and 400 field work orders per month.

What were your objectives?

- Generate a single PDF document including the details of the Work Order and all attached documents.
- Allow the user to use attach documents from wide variety of applications.

Solution

Mainpac EAM stores the attached documents as byte arrays. Based on the file type we were able to use the components (Aspose.Word, Aspose.Cells, Aspose.Imaging etc) from Aspose.Total to convert the byte arrays to PDF memory streams. Microsoft Reporting Services can also provide a PDF stream for the Work Order report. We combine all PDF streams using Aspose.PDF to a single PDF stream that can be sent back to the browser or used to create an email attachment.

Experience

Finding a solution:

We have been using Aspose.PDF to generate PDF Forms and were pleased with the support and documentation provided. When this requirement came up, we visited the Aspose website and found that the Aspose.Total product suite could allow us to convert a wide variety of file formats to PDFs.

We were then able to easily evaluate Aspose.Total by adding the nuget packages into our .Net projects and using the sample code provided on the Aspose website.

Implementation:

When attaching a document, we allowed the user to flag the document as printable.







<div> + - 🔍 📁 </div> <div>50</div>									
<input type="checkbox"/>		Folder	Document	Type	Description	Printable (record)	Printable (global)	Attached On	Source
<input type="checkbox"/>		Crane PM Tasks	Fantuzzi 1000 Hr Gantry Brake Inspection	xlsx	***	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	05/07/21	Attached by Mainpac User
<input type="checkbox"/>		14596	Fantuzzi HV Cable Reel Service	xls	***	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	05/07/21	Attached by Mainpac User
<input type="checkbox"/>			SOH Work Order Service	docx	***	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	05/07/21	Attached by Mainpac User
<input type="checkbox"/>		QuickAttach	Welcome to Word	rtf	***	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	05/07/21	Attached by Mainpac User
<input type="checkbox"/>		713829	STM INVOICE# 4500159767	TXT	***	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	05/07/21	Attached by Mainpac User
<input type="checkbox"/>		Crane Dept. - Parts Photographs	CD-E90051	jpg	***	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	05/07/21	Attached by Mainpac User
Page 1 of 1 (6 items)									

Figure 1. Attach documents to Work Order

The attached documents are saved as byte array in Mainpac EAM.

When a Work Order is issued, we then use the various APIs included in Aspose.Total to convert the byte arrays for the printable documents to a PDF memory streams.

Each Aspose.Total API used very similar code structure to generate a PDF memory stream.

See below example using Aspose.Word to convert DOC, DOCX and RTF documents to PDF Memory Stream.

```

public MemoryStream ConvertDocument(string fileExtension, byte[] originalFileBytes)
{
    if (CanConvert(fileExtension))
    {
        ASPOSELicence.LoadWordsLicence();
        var memoryInStream = new MemoryStream(originalFileBytes);
        Aspose.Words.Document doc = new Aspose.Words.Document(memoryInStream, GetLoadOptions(fileExtension));
        var memoryOutStream = new MemoryStream();
        doc.Save(memoryOutStream, Aspose.Words.SaveFormat.Pdf);
        return memoryOutStream;
    }
    else
        throw new Exception($"{this.GetType().Name} can not be used to convert {fileExtension} files to PDF.");
}

```

See below example using Aspose.Cells to convert Excel XLS and XSLX files to a PDF Memory Stream.

```
public MemoryStream ConvertDocument(string fileExtension, byte[] originalFileBytes )
{
    if (CanConvert(fileExtension))
    {
        ASPOSELicence.LoadCellsLicence();
        var memoryInStream = new MemoryStream(originalFileBytes);
        var loadOptions = GetLoadOptions(fileExtension);
        Aspose.Cells.Workbook workbook = new Aspose.Cells.Workbook(memoryInStream, loadOptions);
        var memoryOutStream = new MemoryStream();
        workbook.Save(memoryOutStream, Aspose.Cells.SaveFormat.Pdf);
        return memoryOutStream;
    }
    else
        throw new Exception($"{this.GetType().Name} can not be used to convert {fileExtension} files to PDF.");
}
```

Once we have the PDF memory streams for each of the attached document and the Reporting Services report, we were able to use the PDFEditor class from Aspose.PDF to merge the memory streams into one PDF document steam, that can be sent back to the browser or used to add an email attachment.

```
public static byte[] Merge(IEnumerable<Stream> inputStreams)
{
    using (MemoryStream pdfStream = new MemoryStream())
    {
        ASPOSELicence.LoadPDFLicence();
        PdfFileEditor pdfEditor = new PdfFileEditor();
        // Concatenate both input MemoryStreams and save to putput MemoryStream
        pdfEditor.Concatenate(inputStreams.ToArray(), pdfStream);
        // Convert MemoryStream back to byte array
        byte[] mergedFile = pdfStream.ToArray();
        return mergedFile;
    }
}
```


After having the infrastructure required by Mainpac EAM to save and retrieve the attached documents, it took less than 2 days to implement the document conversion to PDF using Aspose.Total.

Having the same code structure for the different APIs used by Aspose.Total made the development effort relatively easy. The development team were also able to easily find sample code for each API on the Aspose web site.

Outcome:

Using Aspose.Total we were able to meet our objectives for this enhancement.

We were able to easily add the PDF conversion and merge functionality into our product.

Our customers have been very happy with the functionality, as they have been able to continue to create and store their documents in their original formats, and still produce a single PDF document that can be sent to field technician, overall improving the safety and efficiency of Work Order completion.

Next Steps

We will be looking to incorporate some of the other tools such as barcode generation and email processing provided by Aspose.Total in our product.

Summary

Aspose.Total has met all our expectations. Using the Aspose.Total APIs in code was straight forward and the documentation provided on the Aspose website is excellent. We would recommend Aspose.Total to any other developers looking for a product to convert files to PDFs.