

CASE STUDIES

MiniTool® Software Ltd.

Aspose.Total for .NET Case Study



MiniTool® Software Ltd.

Using Aspose.Total for .NET to fulfil the need for an easy, instant approach to help our users quickly preview files.

19th March 2019.

About MiniTool® Software Ltd.

Since we are working on the new update of [MiniTool Power Data Recovery](#) which deals with the recovery of various types of files, we badly need an easy, instant approach to help our users quickly preview these files before the recovery.

And [Aspose.Total for .NET](#) (hereinafter referred to as Aspose.Total) is such a complete package of all file format APIs (Application Programming Interfaces) for .NET. which seamlessly allows us to use a set of libraries to achieve a wide range of tasks without more service installations.

1 Overview

Aspose.Total includes 13 types of libraries, namely Aspose.CAD.18.3.0, Aspose.Cells.18.8.0, Aspose.Diagram.18.8.0, Aspose.Email.18.7.0, Aspose.EPS.18.8.0, Aspose.Html.18.8.0, Aspose.Imaging.18.7.0, Aspose.Note.18.7.0, Aspose.Pdf.18.1.0, Aspose.PSD.18.8.0, Aspose.Slides.NET.18.7.0, Aspose.Tasks.18.8.0, Aspose.Words.18.8.0. It enables us developers to create, edit, render and transform various popular document formats in any. NET applications.

Pros:

Applying these libraries in our data recovery software comes in the following advantages:

- Avoid installing a bundle of software. For instance, we are able to preview Word, Excel, PPT and do various operations without installing Microsoft's office suite.
- Reduce the cost of software development. We can achieve a variety of application functions without going deep into every file format, huge savings in labor and time costs.
- Get a unified format. Aspose.Total makes it possible for us to convert multiple file formats into a unified file format like PDF for easy outputting and browsing. Such conversions bring easier development.

Cons:

To be honest, we haven't noticed any obvious flaws. If we have to say something, multiple conversions sometimes affect its performance. What's the story?

Since Aspose library does not provide a UI interface, and we also don't want to use other plugins and software to help, we have all file formats converted into TIFF. Such conversion may violate the original intention of Aspose library design, and its performance is greatly reduced after multiple conversions on file format.

In addition, users should pay special attention to the warnings while applying several Aspose libraries simultaneously, because there are several conflicts among these libraries. Be careful when picking up a version of the 13 libraries.

Experience

2 Using the Code

With the help of Aspose.Total, we are able to convert types of files to TIFF for file preview. In this process, some are direct conversion, some need multiple conversions; while for some file formats, conversion is unnecessary.

2.1 Direct Conversion

```
try {
    var doc = new Aspose.Words.Document(fname);
    if (doc.BuiltInDocumentProperties.Pages > maxPageCount) {
        isAllPage = false;
    }
    doc.Save(tiffName, Aspose.Words.SaveFormat.Tiff);
} catch (IndexOutOfRangeException) {
    _isError = true;
    _errorString = _libUnregistered;
} catch (Exception) {
    _isError = true;
    _errorString = _fileBroken;
}
```

The operation of direct conversion is relatively simple as we can just call the Save function to complete this conversion. This operation is in accordance with the design of Aspose.Total and its usage rules. The performance we get totally depends on the performance of the Aspose library itself.

2.2 Multiple Conversions

```
Aspose.Pdf.Document doc;  
bool isColor = true;  
  
if (lowerName.EndsWith(".tex")  
    || lowerName.EndsWith(".xslfo")) {  
    isAllPage = ConvertTextFormat(fname, tiffName);  
    return isAllPage;  
} else if (lowerName.EndsWith(".epub")) {  
    doc = new Aspose.Pdf.Document(fname, new Aspose.Pdf.EpubLoadOptions());  
    isColor = false;  
} else if (lowerName.EndsWith(".cgm")) {  
    doc = new Aspose.Pdf.Document(fname, new Aspose.Pdf.CgmLoadOptions());  
} else if (lowerName.EndsWith(".svg")) {  
    doc = new Aspose.Pdf.Document(fname, new Aspose.Pdf.SvgLoadOptions());  
} else if (lowerName.EndsWith(".xps")) {  
    doc = new Aspose.Pdf.Document(fname, new Aspose.Pdf.XpsLoadOptions());  
    isColor = false;  
} else {  
    doc = new Aspose.Pdf.Document(fname);  
    isColor = false;  
}  
  
if (isColor) {  
    try {  
        doc.Save(pdfName);  
        var pdf = new Aspose.Pdf.Document(pdfName);  
        pdf.Save(pptxName, Aspose.Pdf.SaveFormat.Pptx);  
    } catch (Exception) {  
        m_isError = true;  
        m_errorString = FormatAdapter._fileBroken;  
        return isAllPage;  
    }  
    isAllPage = ConvertSlidesFormat(pptxName, tiffName);  
}
```

In order to maintain the color of CGM and SVG files, we have to save them first as PPT files, and then apply the function of Aspose.Slides.NET library to convert them to TIFF format. This implementation is very inefficient and is contrary to the design of Aspose. Unfortunately, there is no good way to improve the performance of such a conversion.

2.3 No Conversion

It is not a good choice to convert JPEG, JPG, BMP, PNG, and other image file formats. Such files come with colors and the conversion is not necessary.

Summary

Overall, Aspose.Total works great when we have large files and documents to work on. The rich libraries and APIs really impress us and we are inclined to moving to Aspose to make our data recovery software more user-friendly