Aspose Words Case Study

Background

CharityAuctionOrganizer.com provides a platform to allow non-profits to run fundraising auctions. This includes generating a number of documents in support of live events, including: auction catalogs, bidder statements, donor receipts and mailing labels.

Previously these documents were generated as PDFs which were based on a fixed template. Users wanted the ability to customize the appearance of these forms, and over time a large number of specific options where added to set fonts, colors and change alignment and the content that was included.

However, the number of combinations is endless and simply adding more and more options is never going to give users the all flexibility they wanted.

Requirements

What was needed was a way for non-technical users to design the document using a tool they are already comfortable with, and which provides a rich formatting and layout options. These documents could then be uploaded as templates which are filled in with the content hosted in our database.

The most powerful and commonly available option was Microsoft Word. We needed a way to take a Word document supplied by the user and perform the merge server side, generating a file for the user to download.

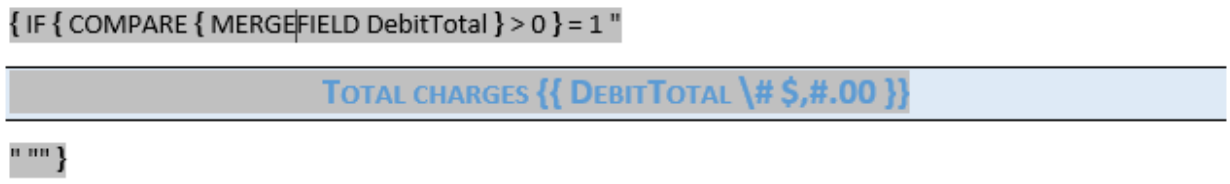
Solution

Aspose Words provides a flexible API for accessing and producing Word documents and has support for invoking the native Word mail merge functionality against a custom data source.

Our first prototype implemented the **IMailMergeDataSource** interface to feed data from the database into the mail merge process. This technique worked well, and was very little code against the Aspose Words API.

However, it had a major drawback: it required that the user implements the template using the Microsoft Word mail merge fields. Whilst mail merge fields work well for simple documents like mailing labels, for more complicated structures they rapidly become unworkable.

For example, a simple conditional block would look like:



This syntax is too complex for our customers to be able to maintain, and very fragile to errors in the formatting.

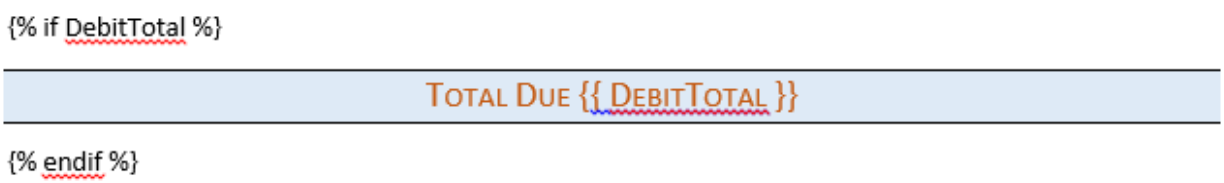
In addition the data model supported by Word mail merge and exposed through IMailMergeDataSource does not support hierarchical data easily. We had many cases of nested data structures which didn't map easily to Word's concept of mail merge regions.

Enhanced Mail Merge

Instead we needed a more flexible solution. The Aspose Words produce already supported a few extensions to the mail merge syntax: the ability to use mustache template syntax for simple field values. We needed to extend this concept to handle more high level flow control primitives.

So rather than using the built-in mail merge features we implemented a custom merge engine via the Aspose API that used control primitives written in the normal flow of the document as text.

The equivalent conditional block using the new syntax is simply:



All the text is simply typed in, there is no need to insert merge fields at all.

In addition to evaluating expressions using mustache templating syntax, there are several control blocks which are implemented:

- {% foreach %} used to iterate over a set of records
- {% row %} used to create rows in a table for each member of a set
- {% sum %} for computing totals by evaluating an expression over a set

In particular the {% row %} primitive makes it very easy to produce tables of content where each row is populated from a different record in the data source. This is a very common requirement of many types of document.

Putting these concepts together leads to templates that look like:

Silent Auction Catalog

{% foreach Categories orderby Name %}

{{ Name }}

Item#		
{% row SilentItems %} {{Number}}	{{Title}} *{{Description}}*	{{Image:Image # Width=80, Collapse=false, Top=6, MarginBottom=6 }}

{% end %}

And which produces output like:

Home and Garden

Item#		
27	**KitchenAid 5 piece stoneware set** *Red colour oven to table bakers nest into each other for easy storage. Extra wide handles. Dishwasher, microwave, freezer and oven safe. Two 6" x 4" bakers, two 9" x 6" bakers, one 9" x 13" baker.*	
10	**Oral B Electric Toothbrush Sonic Complete S-320** *Complete mouth care, professional grade, 3 modes, professional timer, deluxe travel case.*	

The Aspose Words API makes iterating over the structure of the document and manipulating the tree nodes very easy.

Even operations like inserting images into the document require very little code to implement.

Technical Details

The merge engine's first pass needs to convert the document structure into a canonical form where control nodes that are spread over multiple 'run' nodes are collapsed into a single run.

The second pass over the template works by cloning the canonical template, and then looking for the control block runs. These are then evaluated using records from the data model.

For repeating control blocks this requires cloning arbitrary portions of the document, recursively evaluating any control blocks inside them, and inserting the final content back into the document.

Aspose Words makes performing these transforms on the document very straight forward, and provides a consistent navigation model across the various types of content found in the template.