Form Letter Formatting

Intro

Generate a form letter with WordTemplate, then pass it to WordApplication to have formatting applied based on the user's selections.

This demo's template document is in the format of a standard form letter. The recipient and author information is inserted into the document dynamically. You could also write the letter text and image into the document using WordTemplate.

After the form letter is generated with WordTemplate, the document is passed to WordApplication, which then applies formatting (font, font size, etc) based on the user's selections.

Code

```
// This data will be imported
private string recipName = "Bradley Smith";
private string recipStreetAddr = "123 College Street";
private string recipCity = "Boston";
private string recipState = "MA";
private string recipZip = "02201-1020";
private string authName = "John Doyle";
private string authTitle = "Mr.";
// Font specifications that will be applied to the document
private string fontName = "Arial";
private double fontSize = 10;

/// <summary>
/// Build the report with WordTemplate.  Then pass it to WordApplication for
formatting.
/// </summary>
public void GenerateDocument()
{
 // Form the required WordWriter name array.  The elements
 // in this array correspond to the names of the
 // merge fields in the template, and each element's array
 // index should correspond to an element in the value array

 string[] arrNames = {"RecipientName", "RecipientStreetAddr", "RecipientCity",
 "RecipientState", "RecipientZip", "AuthorName", "AuthorTitle"};

 // The value array.  The value of the elements in this array
 // contain info supplied in the web form. These values
 // represent the data that will be dynamically populated
 // in the template file.

 object[] arrValues = {recipName, recipStreetAddr, recipCity,
  recipState, recipZip, authName, authTitle};


 // Create the WordTemplate object
 WordTemplate wt = new WordTemplate();

 // Open the template document
 string templatePath = @"..\..\WordTemplateFiles\FormLetterFormattingTemplate.doc";
 wt.Open(templatePath);
```

```
 // Set the datasource with the name and value arrays defined above
 wt.SetDataSource(arrValues, arrNames);

 // Process the template
 wt.Process();

 // Create the WordApplication object
 WordApplication wwapp = new WordApplication();

 // Open the processed WordTemplate document for formatting
 Document doc = wwapp.Open(wt);

 // Format the document with the selected font and font size
 FormatCharacterRuns(doc);

 // Save the document by streaming it
  // to the client's browser

 wwapp.Save(doc, @"..\..\WordOutputFiles\FormLetterFormatted_out.doc");
}

// Formats all CharacterRuns in the specified Element with the specified font name and
size
private void FormatCharacterRuns(Element e)
{
 // If this element is a CharacterRun, format it
 if (e.ElementType == Element.Type.CharacterRun)
 {
 CharacterRun run = (CharacterRun) e;
 run.Font.FontName = this.fontName;
 run.Font.FontSize = this.fontSize;
 }

 // For each child of this element
 foreach(Element child in e.Children)
 {
 // Call FormatCharacterRuns
 FormatCharacterRuns(child);
 }
```

```
}
```

Downloads

Template: FormLetterFormattingTemplate.doc

Output: FormLetterFormatted_out.doc