Search and Replace

Intro

Demonstrates how to use regular expressions with WordWriter to easily search and replace text.

This sample uses the WordApplication class to open an existing document on the server and perform search and replace actions on it. First, view the stored document to determine which patterns to search for (search terms that work well are *WordWriter*, and *template*). Remember that Regular Expressions are case sensitive. So, to search for "WordWriter" "Wordwriter", use this pattern: Word[Ww]riter.

Code

```
string SearchPattern = "[Ww]ord[Ww]riter";
        string replaceWith = "SoftArtisans";
        string docPath = @"..\..\WordTemplateFiles\SearchReplace.doc";
        /// <summary>
        /// Searches a document for a Regular Expression search term
        /// and replaces all instances of that term with a new string.
        /// The revised document is then saved.
        /// </summary>
        public void SearchDocument()
            /// Open the document you wish to search
            WordApplication wordApp = new WordApplication();
            Document doc = wordApp.Open(docPath);
            // Execute the replacement by specifying the search term
            // as a Regular Expression string, and the replacement string.
            // SearchAndReplace will return the number of replacements made.
            int numReplacements = doc.SearchAndReplace(SearchPattern, replaceWith);
            // If no occurrences of the search pattern were found in the doc, display
            // a message and return.
            if(numReplacements == 0)
                Console.WriteLine("Note: No occurrences of \"{0}\" were found in the
document. " +
                                "Please try another search pattern.", SearchPattern);
                return;
            // Prepend a message to the beginning of the document noting
            // how many replacements were made.
            string text =
                String.Format("Replaced \{0\} instances of \"\{1\}\" with \"\{2\}\"",
                    numReplacements, SearchPattern, replaceWith);
            // Create a new paragraph and insert the note
            Paragraph pg = doc.InsertParagraphBefore(null);
            pg.InsertTextAfter(text, false);
            // Save the edited document
            wordApp.Save(doc, @"..\..\WordOutputFiles\Replaced_out.doc");
    }
```

Downloads

Output: SalesReport_out.doc