docxperiments Documentation

Release 0.1.0

David Seibert

June 02, 2016

Contents

CHAPTER 1

Experiments in OOXML

Contents:

1.1 Introduction

1.2 Materials and Methods

Contents:

1.2.1 Specimen Model

1.2.2 Experiment Design

1.2.3 Operator

1.2.4 Diff Methods

dirdiff.**_standardize**(*nodes*, *parent*)

dirdiff.**get_common**(*a_path*, *b_path*)

dirdiff.**get_left_only**(*a_path*, *b_path*)

dirdiff.**get_right_only**(*a_path*, *b_path*)

dirdiff.**main**()

dirdiff.**walk_dir**(*root*)

class filediff.**ContextDiff**(*a_path*, *b_path*)

> **gen**(*a_lines*, *b_lines*)
>
> **label = 'context_diff'**

class filediff.**Diff**(*a_path*, *b_path*)

> **__init__**(*a_path*, *b_path*)
>
> **_make_report_path**(*exp_dir*, *filename*)

get()

write(*exp_dir*, *filename*)

class filediff.**NDiff**(*a_path*, *b_path*)

gen(*a_lines*, *b_lines*)

label = 'ndiff'

class filediff.**UnifiedDiff**(*a_path*, *b_path*)

gen(*a_lines*, *b_lines*)

label = 'unified_diff'

filediff.**_readlines**(*file_path*)

filediff.**changed**(*a_path*, *b_path*)

filediff.**get_context_diff**(*a_path*, *b_path*)

filediff.**get_diff_battery**(**paths*)

filediff.**get_ndiff**(*a_path*, *b_path*)

filediff.**get_unified_diff**(*a_path*, *b_path*)

filediff.**main**()

1.2.5 Path Utilities

pathutils.**_save_path**(**args*)

pathutils.**data_abs**(*path*)

pathutils.**data_rel**(*path*)

pathutils.**experiments_abs**(*path*)

pathutils.**experiments_rel**(*path*)

pathutils.**ls**(*root*, *depth=None*)

pathutils.**mkdir**(*path*)

pathutils.**mkpath**(**args*)

pathutils.**mkrel**(*path*, *other*)

pathutils.**package_rel**(*path*)

pathutils.**project_rel**(*path*)

pathutils.**specimens_abs**(*path*)

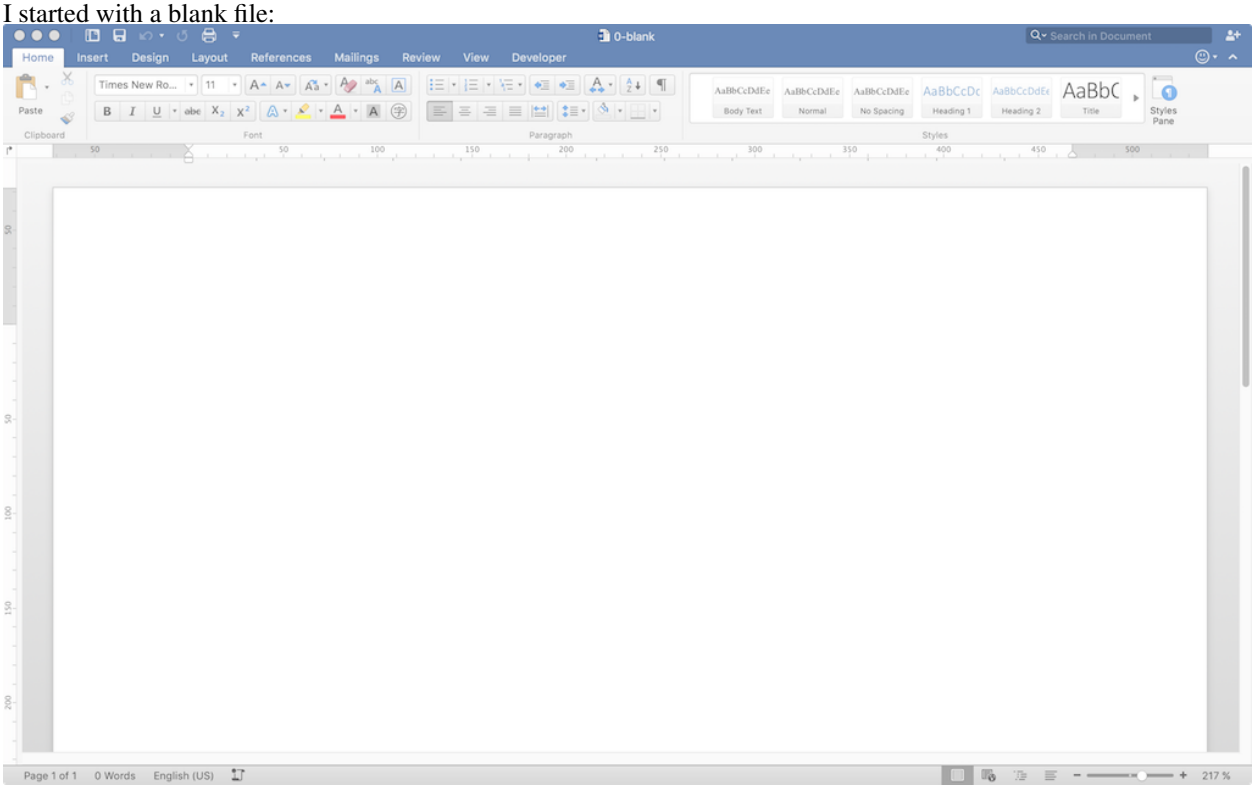
pathutils.**specimens_rel**(*path*)

1.3 Findings

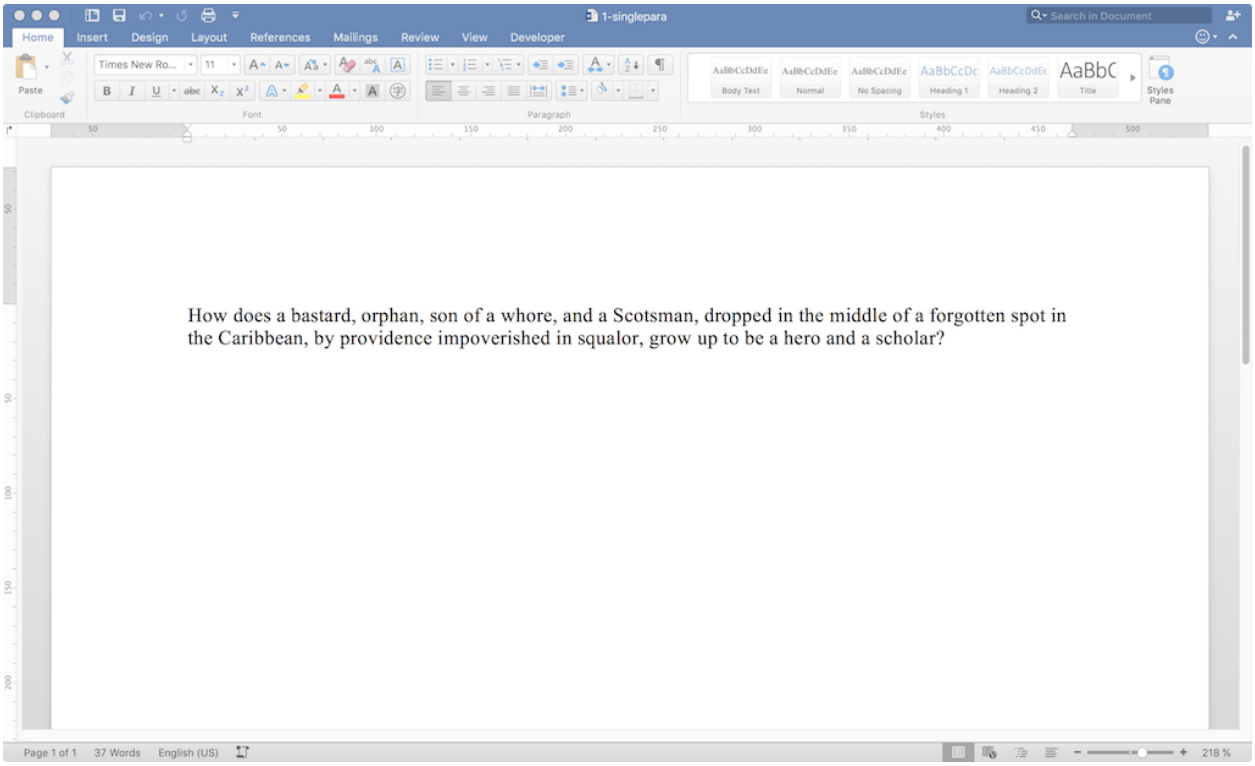
1.3.1 Adding a Single Paragraph

Procedure

I started with a blank file:



And added a single paragraph:



Results

Document XML Changes

The blank `word/document.xml` file has, thankfully, a pretty minimal and manageable structure:

```
<w:body>
 <w:p w14:paraId="2755313D" w14:textId="77777777" w:rsidR="00617040" w:rsidRDefault="00617040">
  <w:bookmarkStart w:id="0" w:name="_GoBack"/>
  <w:bookmarkEnd w:id="0"/>
 </w:p>
 <w:sectPr w:rsidR="00617040" w:rsidSect="00E7316D">
  <w:pgSz w:h="15840" w:w="12240"/>
  <w:pgMar w:bottom="1440" w:footer="720" w:gutter="0" w:header="720" w:left="1440" w:right="1440" w
  <w:cols w:space="720"/>
  <w:docGrid w:linePitch="360"/>
 </w:sectPr>
</w:body>
```

With some paraphrasing and eliding, it's more clear:

```
<body>
 <p>
  <bookmarkStart/>
  <bookmarkEnd/>
 </p>
 <sectionProperty>
  <pageSize/>
  <pageMargin/>
  <cols>
  <docGrid>
```

```
 </sectionProperty>
</body>
```

One paragraph, with some kind of bookmark, and some high-level page layout settings. Cool. Adding one paragraph to the blank produced this (including only the relevant part):

```
<w:p w14:paraId="2755313D" w14:textId="22B9D595" w:rsidR="00617040" w:rsidRDefault="009D3123">
 <w:r w:rsidRPr="009D3123">
  <w:t>
   How does a bastard, orphan, son of a whore, and a Scotsman, dropped in the middle of a forgotten s
  </w:t>
 </w:r>
 <w:bookmarkStart w:id="0" w:name="_GoBack"/>
 <w:bookmarkEnd w:id="0"/>
</w:p>
```

As paraphrased:

```
<p>
 <r>
  <t>
    How does a bastard, orphan, son of a whore,
    and a Scotsman, dropped in the middle of
    a forgotten spot in the Caribbean,
    by providence impoverished in squalor,
    grow up to be a hero and a scholar?
  </t>
 </r>
 <bookmarkStart/>
 <bookmarkEnd/>
</p>
```

Now there's a *run* and a *text* tag surrounding the text I entered. Nothing else get_changed.

Other Changes

In `word/settings.xml` there are two new `<w:rsid>` tags, inserted with the others alphabetically:

- `<w:rsid w:val="009D3123"/>`
- `<w:rsid w:val="00FD3E79"/>`

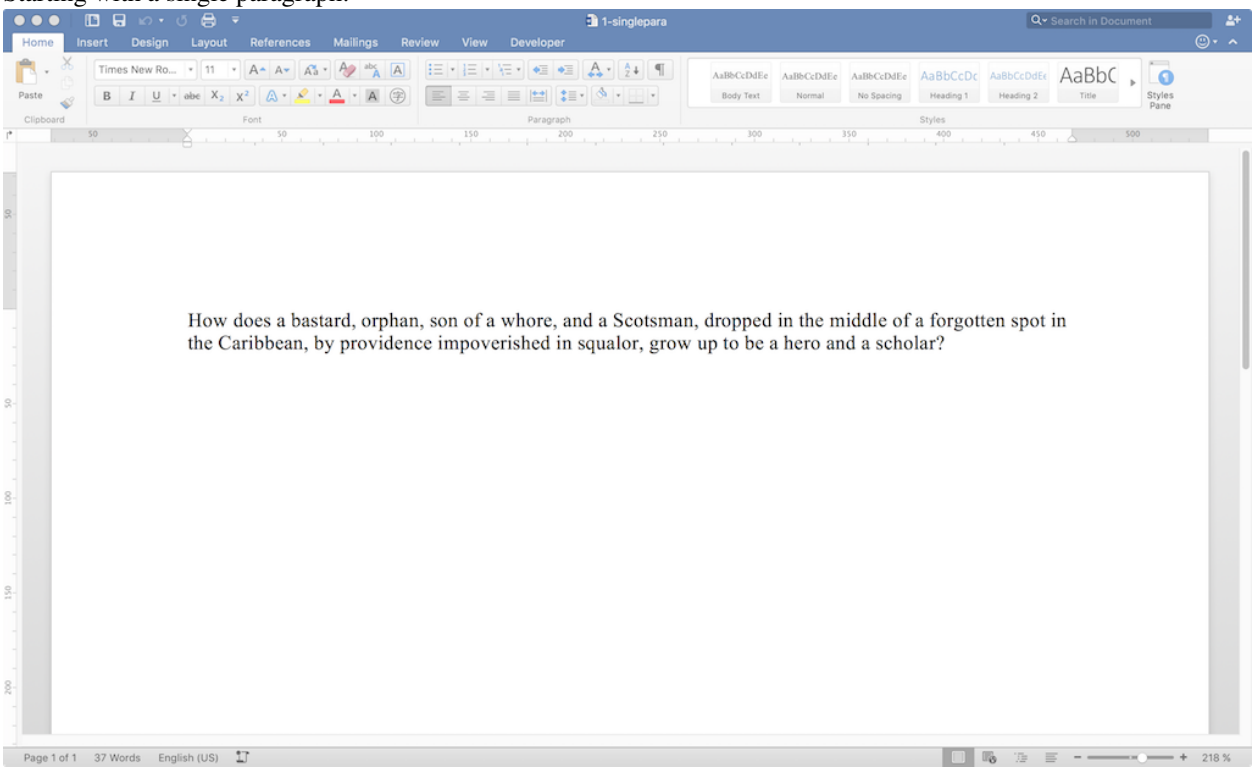
In `docProps/app.xml` a few counts were modified for the new content:

- `TotalTime` from 0 to 1
- `Words` from 0 to 29
- `Characters` from 0 to 166
- `Lines` from 0 to 1
- `Paragraphs` from 0 to 1
- `CharacterWithSpaces` from 0 to 194

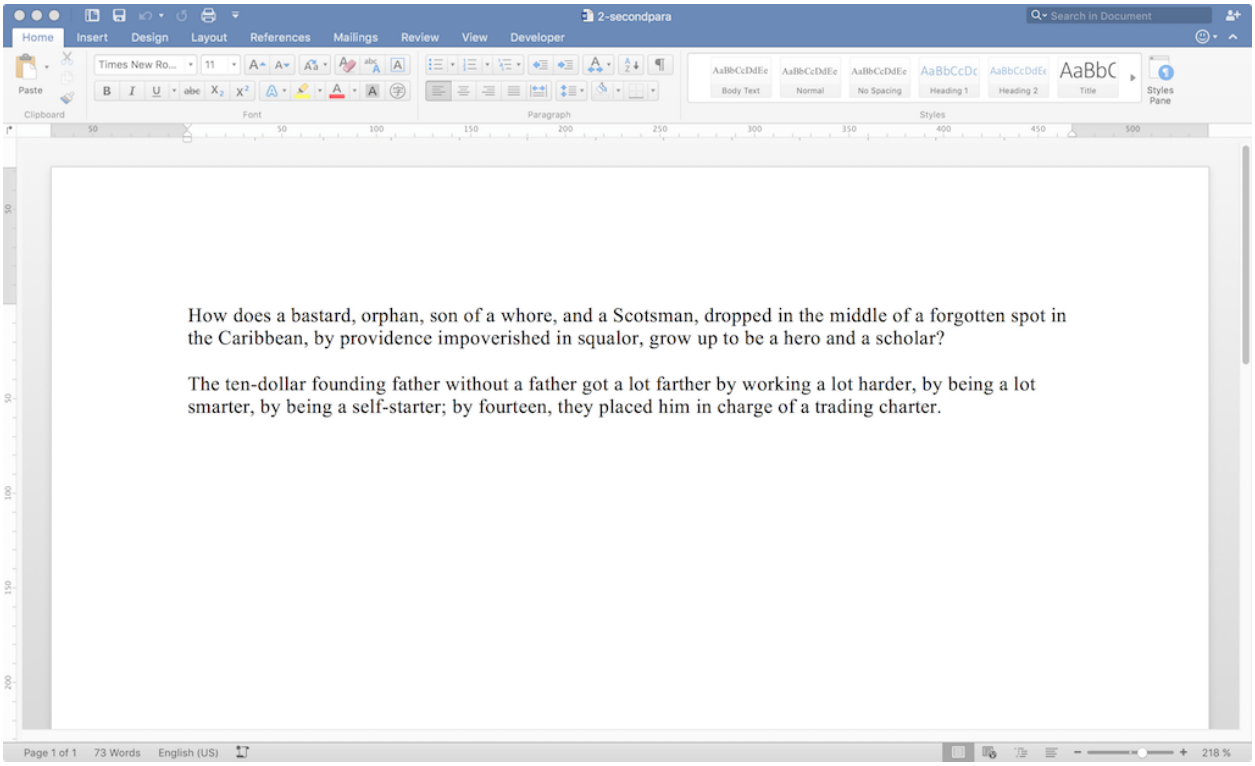
1.3.2 Adding a Second Paragraph

Procedure

Starting with a single paragraph:



I added a second paragraph:



Results

Document XML Changes

Something went wrong. Adding the first paragraph gave me this:

```
<body>
 <p>
  <r>
   <t>
    How does a bastard, orphan, son of a whore,
    and a Scotsman, dropped in the middle of
    a forgotten spot in the Caribbean,
    by providence impoverished in squalor,
    grow up to be a hero and a scholar?
   </t>
  </r>
  <bookmarkStart/>
  <bookmarkEnd/>
 </p>
 <sectionProperty>
  <pageSize/>
  <pageMargin/>
  <cols>
  <docGrid>
 </sectionProperty>
</body>
```

Somehow, the second paragraph yielded *this*:

```
<body>
 <p>
  <r>
   <t>
    How does a bastard, orphan, son of a whore,
    and a Scotsman, dropped in the middle of
    a forgotten spot in the Caribbean,
    by providence impoverished in squalor,
    grow up to be a hero and a scholar?
   </t>
  </r>
 </p>
 <p>
  <r>
   <t>
    The ten-dollar founding father without a father
    got a lot farther by working a lot harder,
    by being a lot smarter, by being a self-starter;
    by fourteen, they placed him
   </t>
  </r>
  <r>
   <t>
    in charge of a trading charter.
   </t>
  </r>
  <bookmarkStart/>
  <bookmarkEnd/>
 </p>
 <sectionProperty>
  <pageSize/>
  <pageMargin/>
  <cols>
  <docGrid>
 </sectionProperty>
</body>
```

What happened? Somehow, the second paragraph got divided into a pair of runs. I recreated the docx and the same thing happened, in the same location. I wonder what it means.

Other Changes

Another `<w:rsid>` in `word/settings.xml`: `<w:rsid w:val="00B61498"/>`

What does it mean?

The `<sectionProperty>` tag changed: the `<w:rsidR>` attribute changed from "00617040" to "00B61498"... Which is also the number from just above...

The `docProps/app.xml` counts changed again:

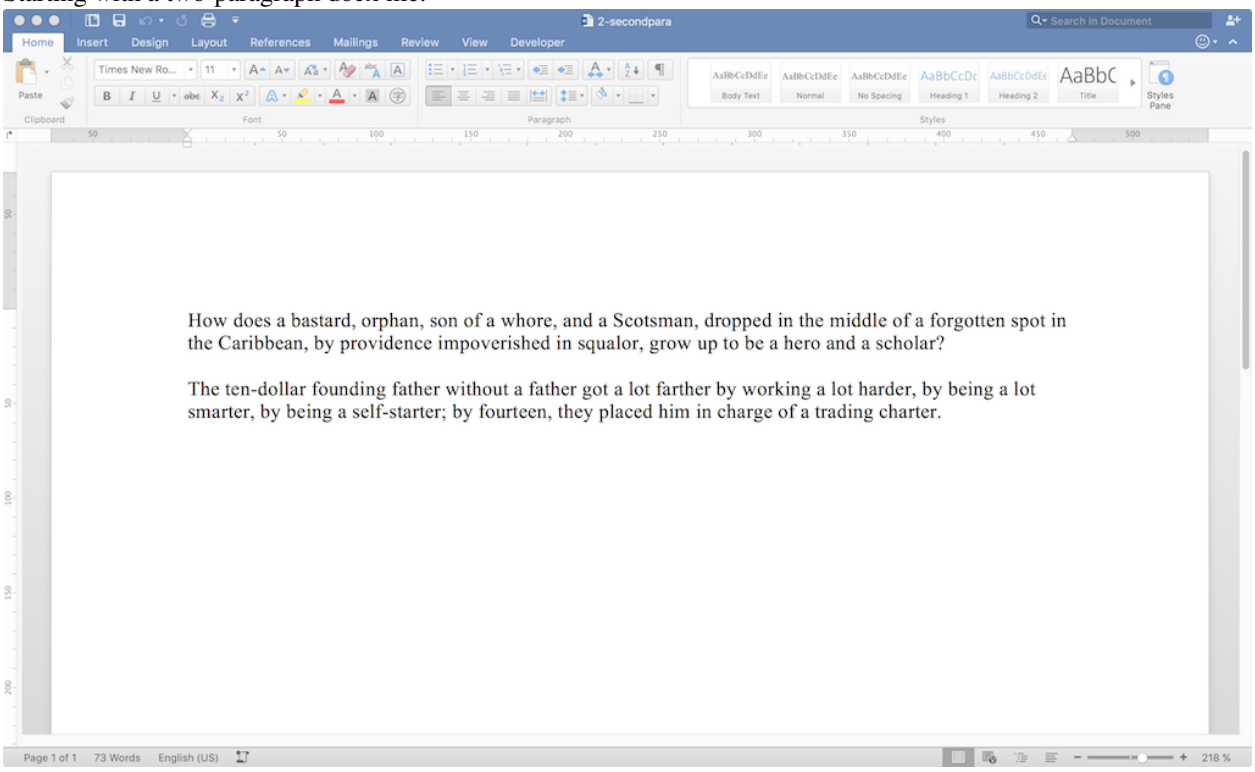
- `Words` from 29 to 59
- `Characters` from 166 to 337
- `Lines` from 1 to 2
- `CharacterWithSpaces` from 194 to 395

`Paragraphs`...did not change...?! Still 1. Maybe because I didn't put a linefeed at the end of the second paragraph?

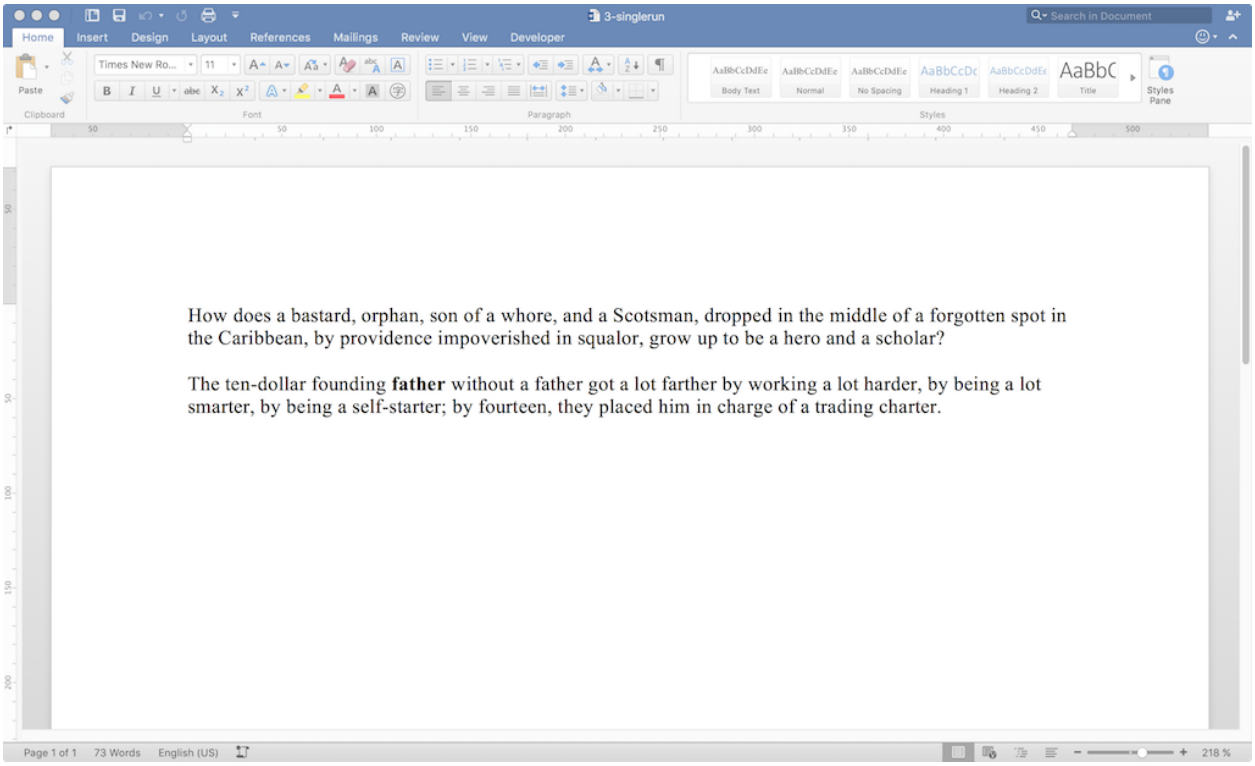
1.3.3 Adding a Run

Procedure

Starting with a two-paragraph docx file:



I made one word bold:



Results

Document XML Changes

We started with this:

```
<body>
 <p>
  <r>
   <t>
    How does a bastard, orphan, son of a whore,
    and a Scotsman, dropped in the middle of
    a forgotten spot in the Caribbean,
    by providence impoverished in squalor,
    grow up to be a hero and a scholar?
   </t>
  </r>
 </p>
 <p>
  <r>
   <t>
    The ten-dollar founding father without a father
    got a lot farther by working a lot harder,
    by being a lot smarter, by being a self-starter;
    by fourteen, they placed him
   </t>
  </r>
  <r>
   <t>
    in charge of a trading charter.
```

```
   </t>
  </r>
  <bookmarkStart/>
  <bookmarkEnd/>
 </p>
 <sectionProperty>
  <pageSize/>
  <pageMargin/>
  <cols>
  <docGrid>
 </sectionProperty>
</body>
```

With the new run, we have this:

```
<body>
 <p>
  <r>
   <t>
    How does a bastard, orphan, son of a whore,
    and a Scotsman, dropped in the middle of
    a forgotten spot in the Caribbean,
    by providence impoverished in squalor,
    grow up to be a hero and a scholar?
   </t>
  </r>
 </p>
 <p>
  <r>
   <t>
    The ten-dollar founding
   </t>
  </r>
  <r>
   <rProperty>
    <b/>
   </rProperty>
   <t>
    father
   </t>
  </r>
  <r>
   <t>
    witho
   </t>
  </r>
  <bookmarkStart/>
  <bookmarkEnd/>
  <r>
   <t>
    ut a father got a lot farther
    by working a lot harder,
    by being a lot smarter,
    by being a self-starter;
    by fourteen, they placed him
   </t>
  </r>
  <r>
   <t>
```

```
    in charge of a trading charter.
   </t>
  </r>
 </p>
 <sectionProperty>
  <pageSize/>
  <pageMargin/>
  <cols>
  <docGrid>
 </sectionProperty>
</body>
```

So, here's what happened:

- The `<bookmark/>` tags moved up.
- The `<sectionProperty>`'s `rsidR` attribute changed again from "00B61498" to "00F564C0".
- The `rsidRPr` attribute for the run surrounding the second paragraph changed from "00B61498" to "00F564C0", which is the same as for the run that starts with "witho" and "ut a father got...".
- The attributes for the first paragraph all changed:

```
-    <w:p w14:paraId="6657B44B" w14:textId="24C88B5A" w:rsidR="00B61498" w:rsidRDefault="00B61498
+    <w:p w14:paraId="60C4CAE9" w14:textId="6037BB23" w:rsidR="00F564C0" w:rsidRDefault="00F564C0
```

Other Changes

Lost one `<w:rsid>` and gained gained two:

```
-   <w:rsid w:val="00B61498"/>
+   <w:rsid w:val="00E326CC"/>
+   <w:rsid w:val="00F564C0"/>
```

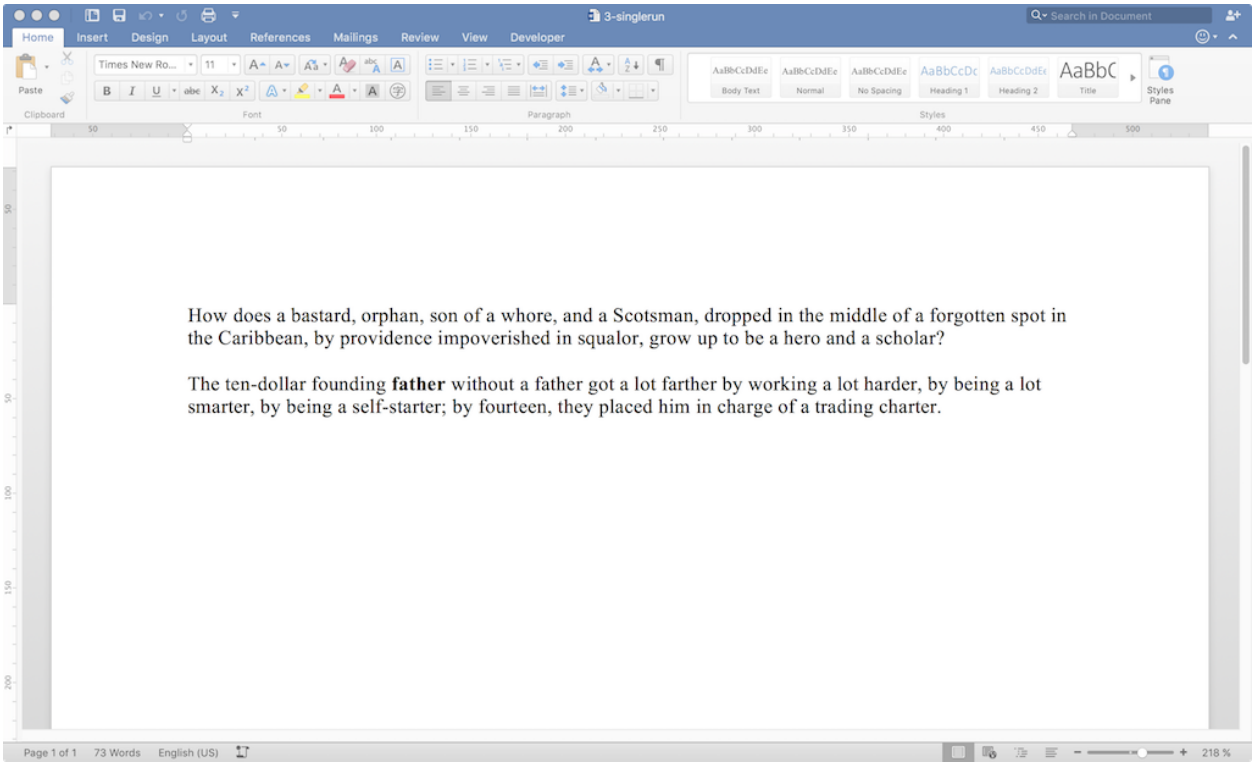
No changes in the `docProps/app.xml` counts:

- `Words` is the same, 59
- `Characters` is the same, 337
- `Lines` is the same, 2
- `CharacterWithSpaces` is the same, 395
- `Paragraphs` is the same, 1...

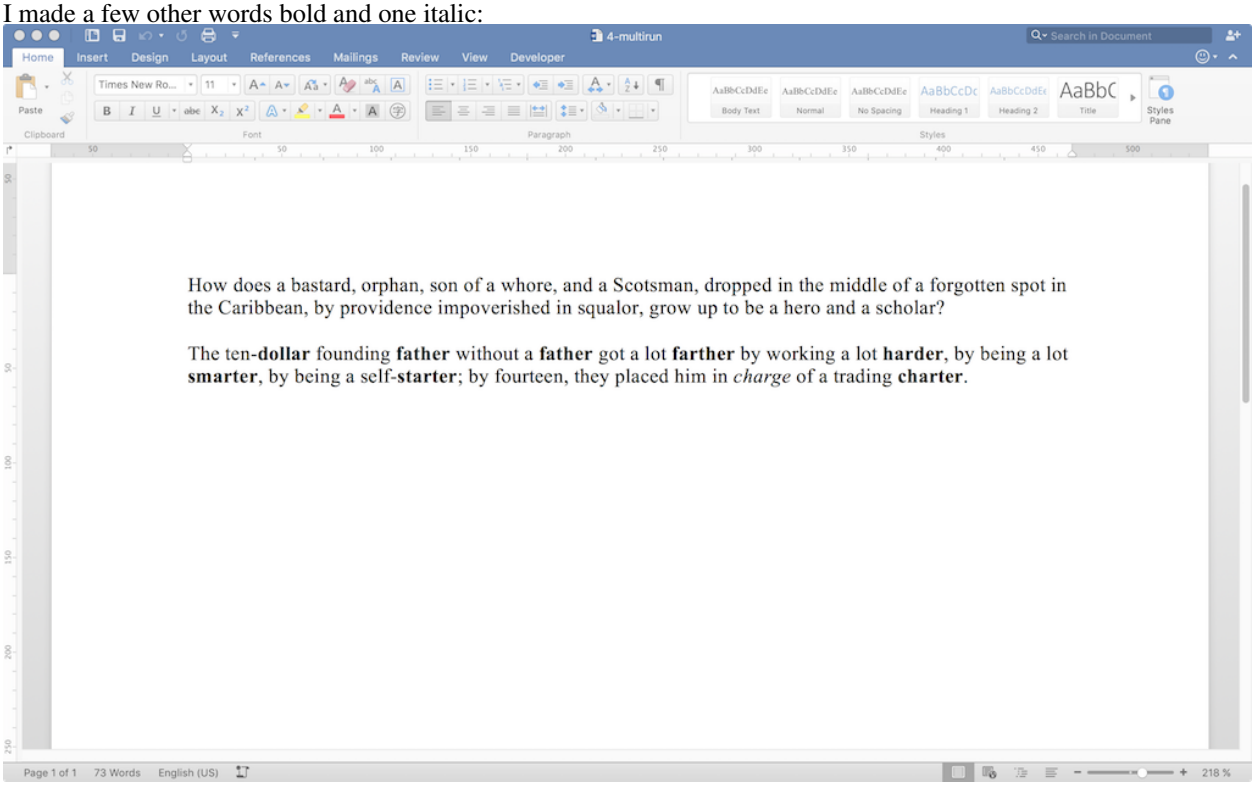
1.3.4 Adding Multiple Runs

Procedure

Starting with a two-paragraph docx file with one bold word:



I made a few other words bold and one italic:



Results

Document XML Changes

Strap in. Starting with one run:

```
<body>
 <p>
  <run>
    How does a bastard, orphan, son of a whore,
    and a Scotsman, dropped in the middle of
    a forgotten spot in the Caribbean,
    by providence impoverished in squalor,
    grow up to be a hero and a scholar?
 </p>
 <p>
  <run>
    The ten-dollar founding
  <r>
   <rProperty>
    <b/>
   </rProperty>
   <t>
    father
   </t>
  </r>
  <r>
   <t>
    witho
   </t>
  </r>
  <bookmarkStart/>
  <bookmarkEnd/>
  <r>
   <t>
    ut a father got a lot farther
    by working a lot harder,
    by being a lot smarter,
    by being a self-starter;
    by fourteen, they placed him
   </t>
  </r>
  <r>
   <t>
    in charge of a trading charter.
   </t>
  </r>
 </p>
 <sectionProperty>
  <pageSize/>
  <pageMargin/>
  <cols>
  <docGrid>
 </sectionProperty>
</body>
```

We now have a seemingly endless amount:

```
<body>
 <p>
  <run>
    How does a bastard, orphan, son of a whore,
    and a Scotsman, dropped in the middle of
    a forgotten spot in the Caribbean,
    by providence impoverished in squalor,
    grow up to be a hero and a scholar?

 <p>
  <run> The ten-                          <!-- 1 -->
  <boldrun> dollar                        <!-- 2 -->
  <run> founding                          <!-- 1 -->
  <boldrun> father                        <!-- 3 -->
  <run> without a                         <!-- 1 -->
  <boldrun> father                        <!-- 2 -->
  <run> got a                             <!-- 1 -->
  <bookmark/>
  <run> lot                               <!-- 1 -->
  <boldrun> farther                       <!-- 2 -->
  <run> by working a lot                  <!-- 1 -->
  <boldrun> harder                        <!-- 2 -->
  <run> , by being a lot                  <!-- 1 -->
  <boldrun> smarter                       <!-- 2 -->
  <run> , by being a self-                <!-- 1 -->
  <boldrun> starter                       <!-- 2 -->
  <run> ; by fourteen, they place him     <!-- 1 -->
  <run> in
  <italicrun> charge                      <!-- 2 -->
  <run> of a trading
  <boldrun> charter                       <!-- 2 -->
  <run> .

 <sectionProperty>
  <pageSize/>
  <pageMargin/>
  <cols>
  <docGrid>
 </sectionProperty>
</body>
```

Runs.

Other Changes

Gained a <w:rsid>:

```
+  <w:rsid w:val="008937FE"/>
```

1.4 Spin-off: Minimum Viable Markup

1.4.1 Intro

I was playing around with the innards of some docx files and found that all this `rsid` business was *quite complicated*. Intimidatingly so.

However, I found that adding in a few run, text, and bold tags to a file and recompressing it did work *even without any of those tags.* From there I messed about further and eventually I even copied in some raw markup from a fairly complicated table in another docx file, with custom paragraph styles, and that worked! The styles were stripped, of course, but the table made it through intact, with the correct proportions and borders, with no complaints from Word. I didn't edit any of the markup; it had undefined style names and undefined document property fields.

So, I thought: How much can I get away with? How strict *is* word?

Recompressing my pretty files does *not* work, which is too bad. Editing the ugly files is annoying. I think the extra spaces between the tags is the problem, but I'm not going to try to solve that for now. I started removing tags at random from some of the ugly files and found that I could get away with not having many of them. That means that, theoretically, creating a docx from scratch might be manageable if I can just need to nail down the bare minimum that Word requires. After tooling around randomly, I thought I should make it a bit more rigorous. Starting with the docx file from specimen four, I'm going to strip away as much as I can.

Here we go.

The whole directory structure looks like this:

```
DOCX
-- [Content_Types].xml
-- _rels
|   -- .rels
-- docProps
|   -- app.xml
|   -- core.xml
|   -- thumbnail.jpeg
-- word
    -- _rels
    |   -- document.xml.rels
    -- document.xml
    -- fontTable.xml
    -- settings.xml
    -- styles.xml
    -- theme
    |   -- theme1.xml
    -- webSettings.xml
```

It's pretty scary. But, a lot of this can be deleted without a second thought, it turns out. This is the most minimal structure I could get to work:

```
DOCX
-- [Content_Types].xml
-- _rels
|   -- .rels
-- word
    -- document.xml
    -- theme
        -- theme1.xml
```

word/document.xml

This, obviously, is the main piece of the package. The original looks like this, in essence:

```
<?xml version="1.0" encoding="utf-8"?>
<w:document mc:Ignorable="w14 w15 wp14"
            xmlns:m="[schema url]"
            xmlns:mc="[schema url]"
            xmlns:mo="[schema url]"
            xmlns:mv="[schema urn]"
            xmlns:o="[schema urn]"
            xmlns:r="[schema url]"
            xmlns:v="[schema urn]"
            xmlns:w="[schema url]"
            xmlns:w10="[schema urn]"
            xmlns:w14="[schema url]"
            xmlns:w15="[schema url]"
            xmlns:wne="[schema url]"
            xmlns:wp="[schema url]"
            xmlns:wp14="[schema url]"
            xmlns:wpc="[schema url]"
            xmlns:wpg="[schema url]"
            xmlns:wpi="[schema url]"
            xmlns:wps="[schema url]">
 <w:body>
  <w:p  w14:paraId="2755313D"
        w14:textId="22B9D595"
        w:rsidR="00617040"
        w:rsidRDefault="009D3123">
   <w:r w:rsidRPr="009D3123">
    <w:t>
     How does a bastard, orphan, son of a whore,
     and a Scotsman, dropped in the middle of
     a forgotten spot in the Caribbean,
     by providence impoverished in squalor,
     grow up to be a hero and a scholar?
    </w:t>
   </w:r>
  </w:p>
  <w:p  w14:paraId="60C4CAE9"
        w14:textId="6037BB23"
        w:rsidR="00F564C0"
        w:rsidRDefault="00F564C0">
   <w:r w:rsidRPr="00F564C0">
    <w:t>
     The ten-
    </w:t>
   </w:r>
   <w:r w:rsidRPr="008937FE">
    <w:rPr>
     <w:b/>
    </w:rPr>
    <w:t>
     dollar
    </w:t>
   </w:r>
   <w:r w:rsidRPr="00F564C0">
    <w:t xml:space="preserve">
     founding
```

```
 </w:t>
</w:r>
<w:r w:rsidRPr="00E326CC">
 <w:rPr>
  <w:b/>
 </w:rPr>
 <w:t>
  father
 </w:t>
</w:r>
<w:r w:rsidRPr="00F564C0">
 <w:t xml:space="preserve">
  without a
 </w:t>
</w:r>
<w:r w:rsidRPr="008937FE">
 <w:rPr>
  <w:b/>
 </w:rPr>
 <w:t>
  father
 </w:t>
</w:r>
<w:r w:rsidRPr="00F564C0">
 <w:t xml:space="preserve">
  got a
 </w:t>
</w:r>
<w:bookmarkStart w:id="0"
                 w:name="_GoBack"/>
<w:bookmarkEnd w:id="0"/>
<w:r w:rsidRPr="00F564C0">
 <w:t xml:space="preserve">
  lot
 </w:t>
</w:r>
<w:r w:rsidRPr="008937FE">
 <w:rPr>
  <w:b/>
 </w:rPr>
 <w:t>
  farther
 </w:t>
</w:r>
<w:r w:rsidRPr="00F564C0">
 <w:t xml:space="preserve">
  by working a lot
 </w:t>
</w:r>
<w:r w:rsidRPr="008937FE">
 <w:rPr>
  <w:b/>
 </w:rPr>
 <w:t>
  harder
 </w:t>
</w:r>
<w:r w:rsidRPr="00F564C0">
```

```
    <w:t xml:space="preserve">
     , by being a lot
    </w:t>
   </w:r>
   <w:r w:rsidRPr="008937FE">
    <w:rPr>
     <w:b/>
    </w:rPr>
    <w:t>
     smarter
    </w:t>
   </w:r>
   <w:r w:rsidRPr="00F564C0">
    <w:t>
     , by being a self-
    </w:t>
   </w:r>
   <w:r w:rsidRPr="008937FE">
    <w:rPr>
     <w:b/>
    </w:rPr>
    <w:t>
     starter
    </w:t>
   </w:r>
   <w:r w:rsidRPr="00F564C0">
    <w:t xml:space="preserve">
     ; by fourteen, they placed him
    </w:t>
   </w:r>
   <w:r>
    <w:t xml:space="preserve">
     in
    </w:t>
   </w:r>
   <w:r w:rsidRPr="008937FE">
    <w:rPr>
     <w:i/>
    </w:rPr>
    <w:t>
     charge
    </w:t>
   </w:r>
   <w:r>
    <w:t xml:space="preserve">
     of a trading
    </w:t>
   </w:r>
   <w:r w:rsidRPr="008937FE">
    <w:rPr>
     <w:b/>
    </w:rPr>
    <w:t>
     charter
    </w:t>
   </w:r>
   <w:r>
    <w:t>
```

```
      .
     </w:t>
    </w:r>
   </w:p>
   <w:sectPr w:rsidR="00F564C0"
             w:rsidSect="00E7316D">
    <w:pgSz  w:h="15840"
             w:w="12240"/>
    <w:pgMar w:bottom="1440"
             w:footer="720"
             w:gutter="0"
             w:header="720"
             w:left="1440"
             w:right="1440"
             w:top="1440"/>
    <w:cols w:space="720"/>
    <w:docGrid w:linePitch="360"/>
   </w:sectPr>
  </w:body>
</w:document>
```

And this is a two-paragraph document! This does not bode well. Let's see what simplifications are available:

1. The `w:rsidR` tags can be removed.
2. The `w:rsidRPr` tags can be removed.
3. The `w14:textID` and `w14:paraID` tags can be removed.
4. The `w:rsidDefault` tags can be removed.
5. The `w:rsidSect` tag can be removed.
6. The `w:bookmarkStart` and `w:bookmarkEnd` tags can be removed.

That leaves us with the following. I've taken out the `w:` prefix on everything, the `w:document` attributes, and I collapsed the run tags:

```
<?xml version="1.0" encoding="utf-8"?>
<document>
 <body>
  <p>   <!-- paragraph #1 -->
   <r><t>   <!-- regular -->
     How does a bastard, orphan, son of a whore,
     and a Scotsman, dropped in the middle of a
     forgotten spot in the Caribbean, by providence
     impoverished in squalor, grow up to be a hero
     and a scholar?
    </t></r>
  </p>
  <p>   <!-- paragraph #2 -->
   <r><t>   <!-- regular -->
     The ten-
    </t></r>
   <r><rPr><b/></rPr><t>    <!-- bold -->
     dollar
    </t></r>
   <r><t xml:space="preserve">  <!-- regular -->
     founding
    </t></r>
   <r><rPr><b/></rPr><t>    <!-- bold -->
```

```
    father
   </t></r>
  <r><t xml:space="preserve">  <!-- regular -->
    without a
   </t></r>
  <r><rPr><b/></rPr><t>    <!-- bold -->
    father
   </t></r>
  <r><t xml:space="preserve">  <!-- regular -->
    got a
   </t></r>
  <r><t xml:space="preserve">  <!-- regular -->
    lot
   </t></r>
  <r><rPr><b/></rPr><t>    <!-- bold -->
    farther
   </t></r>
  <r><t xml:space="preserve">  <!-- regular -->
    by working a lot
   </t></r>
  <r><rPr><b/></rPr><t>    <!-- bold -->
    harder
   </t></r>
  <r><t xml:space="preserve">  <!-- regular -->
    , by being a lot
   </t></r>
  <r><rPr><b/></rPr><t>    <!-- bold -->
    smarter
   </t></r>
  <r><t>   <!-- regular -->
    , by being a self-
   </t></r>
  <r><rPr><b/></rPr><t>    <!-- bold -->
    starter
   </t></r>
  <r><t xml:space="preserve">  <!-- regular -->
    ; by fourteen, they placed him
   </t></r>
  <r><t xml:space="preserve">  <!-- regular -->
    in
   </t></r>
  <r><rPr><i/></rPr><t>    <!-- italic -->
    charge
   </t></r>
  <r><t xml:space="preserve">  <!-- regular -->
    of a trading
   </t></r>
  <r><rPr><b/></rPr><t>    <!-- bold -->
    charter
   </t></r>
  <r><t>   <!-- regular -->
    .
   </t></r>
 </p>
 <sectPr>
  <pgSz    h="15840"
           w="12240"/>
  <pgMar   bottom="1440"
```

```
            footer="720"
            gutter="0"
            header="720"
            left="1440"
            right="1440"
            top="1440"/>
   <cols space="720"/>
   <docGrid linePitch="360"/>
  </sectPr>
 </body>
</document>
```

One more thing in terms of modifying document.xml itself: A few of these runs don't seem to be necessary:

```
<r><t xml:space="preserve">  <!-- regular -->
  got a
 </t></r>
<r><t xml:space="preserve">  <!-- regular -->
  lot
 </t></r>
 . . .
<r><t xml:space="preserve">  <!-- regular -->
  ; by fourteen, they placed him
 </t></r>
<r><t xml:space="preserve">  <!-- regular -->
  in
 </t></r>
```

Done:

```
<r><t xml:space="preserve">  <!-- regular -->
  got a lot
 </t></r>
 . . .
<r><t xml:space="preserve">  <!-- regular -->
  ; by fourteen, they placed him in
 </t></r>
```

So, by this point the document has become fairly simple. Most of the complication is due to the bizarre OOXML markup scheme. What is the point of the `<t>` tag? I think we can get rid of them. We can get rid of a *lot* of this mess.

New Markup

I'll just make a few substitutions:

`<t xml:space="preserve">` –> Nothing.

The obvious point is to mark the elements where spacing is literal. It seems simple to include a rule in a parser that says any multi-word element preserves space.

`<r><t>:` –> `<n>`

This is just normal, unstyled text. I don't know why it needs markup. To be be conservative, I'll give it the tag 'n' for 'normal'.

`<r><rPr><b/></rPr><t>` –> `<b>`

Because obviously.

`<r><rPr><i/></rPr><t>` –> `<i>`

Because obviously.

Leaving aside the section properties for now, here is the new markup:

```
<docx>
 <body>
  <p><n>How does a bastard, orphan, son of a whore,
     and a Scotsman, dropped in the middle of a
     forgotten spot in the Caribbean, by providence
     impoverished in squalor, grow up to be a hero
     and a scholar?</n></p>
  <p><n>The ten-</n><b>dollar</b><n> founding </n>
   <b>father</b><n> without a </n><b>father</b>
   <n> got a lot </n><b>farther</b><n> by working a lot </n>
   <b>harder</b><n>, by being a lot </n><b>smarter</b>
   <n>, by being a self-</n><b>starter</b><n>; by fourteen,
   they placed him in </n><i>charge</i><n> of a trading </n>
   <b>charter</b><n>.</n></p>
 </body>
 <sectionProperties/>
</docx>
```

Not bad. Let's build ourselves a compiler.

```
>>> import os, difflib
>>> from bs4 import BeautifulSoup
>>> simplified_doc_xml = \
...     os.path.realpath(
...         '../../../synthesis/stages/'
...         '13.8-documentxml_redundant_runs_removed'
...         '/decomposed/word/document.xml')
>>> os.path.exists(simplified_doc_xml)
True
>>> with open(simplified_doc_xml) as f:
...     target_markup = f.read()
>>> print target_markup
<?xml version="1.0"...
>>> with open('docx_boilerplate.xml', 'r') as f:
...     docx_boilerplate = f.read()[:-1]
>>> print docx_boilerplate
<?xml version="1.0"...
>>> with open('section_properties.xml', 'r') as f:
...     section_properties = f.read()
>>> print section_properties
<w:sectPr ><w:pgSz w:w="12240"...
>>> with open('test_markup.xml', 'r') as f:
...     test_markup = f.read()
>>> print test_markup
<docx><body><p><n>How does a bastard...
>>> replacements = [
...     ('<n>', '<w:r><w:t xml:space="preserve">'),
...     ('<b>', '<w:r><w:rPr><w:b/></w:rPr><w:t>'),
...     ('<i>', '<w:r><w:rPr><w:i/></w:rPr><w:t>'),
...     ('</n>', '</w:t></w:r>'),
...     ('</b>', '</w:t></w:r>'),
...     ('</i>', '</w:t></w:r>'),
...     ('<p>', '<w:p>'),
...     ('</p>', '</w:p>'),
...     ('<sectionProperties/>', section_properties),
...     ('<body>', '<w:body>'),
```

```
...     ('</body>', '</w:body>'),
...     ('<docx>', docx_boilerplate),
...     ('</docx>', '</w:document>'),
...     ]
>>> intermediate = test_markup
>>> for i, j in replacements:
...     intermediate = intermediate.replace(i, j)
>>> with open('output.xml', 'w') as f:
...     f.write(intermediate)
>>> pretty_test_markup = \
...     BeautifulSoup(intermediate, "xml").prettify()
>>> pretty_target_markup = \
...     BeautifulSoup(target_markup, "xml").prettify()
>>> diff = difflib.unified_diff(
...     pretty_target_markup.split("\n"),
...     pretty_test_markup.split("\n")
...     )
>>> print '\n'.join([ line for line in diff])
---

+++

@@ -3,14 +3,14 @@

  <w:body>
   <w:p>
    <w:r>
-    <w:t>
+    <w:t xml:space="preserve">
      How does a bastard, orphan, son of a whore,
      and a Scotsman, dropped in the middle of a
      forgotten spot in the Caribbean,
      by providence impoverished in squalor,
      grow up to be a hero and a scholar?
     </w:t>
    </w:r>
   </w:p>
   <w:p>
    <w:r>
-    <w:t>
+    <w:t xml:space="preserve">
      The ten-
     </w:t>
    </w:r>
@@ -88,7 +88,7 @@

     </w:t>
    </w:r>
    <w:r>
-    <w:t>
+    <w:t xml:space="preserve">
      , by being a self-
     </w:t>
    </w:r>
@@ -127,7 +127,7 @@

     </w:t>
    </w:r>
```

```
    <w:r>
-    <w:t>
+    <w:t xml:space="preserve">
      .
     </w:t>
    </w:r>
```

Obviously, I need to touch up the preserve spacing, but pop the output into the archive and you've got yourself a working docx compiler. Huzzah!

Minor Files

word/styles.xml

Deleted and opened without complaint, but the styles were all different, naturally. I'll have to come back to this one.

word/theme/theme1.xml

After deleting it, Word opened the file but had to repair it. I looked inside it and it was pretty uninteresting.

_rels/.rels

Initial State The original file looks like this, except that I've prettified it:

```
<?xml version="1.0" encoding="utf-8"?>
<Relationships xmlns="[schema url]">
 <Relationship  Id="rId3"
                Target="docProps/core.xml"
                Type="[schema url]">
 <Relationship  Id="rId4"
                Target="docProps/app.xml"
                Type="[schema url]">
 <Relationship  Id="rId1"
                Target="word/document.xml"
                Type="[schema url]">
 <Relationship  Id="rId2"
                Target="docProps/thumbnail.jpeg"
                Type="[schema url]">
</Relationships>
```

Pretty verbose. I'll simplify it to the following for clarity's sake:

```
<?xml?>
<Rels>
 <Rel Target="docProps_core.xml"/>
 <Rel Target="docProps_app.xml"/>
 <Rel Target="word_document.xml"/>
 <Rel Target="docProps_thumbnail.jpeg"/>
</Rels>
```

Simplification

1. Removing the file entirely: FAIL

2. Removing all four `<Relationship/>` tags: FAIL
3. Removing the `docProps/core.xml` tag: SUCCESS
4. Removing the `docProps/app.xml` tag: SUCCESS
5. Removing the `docProps/thumbnail.xml` tag: SUCCESS

The last three steps were cumulative. In the end, I had a working file with this:

```
<?xml?>
<Rels>
 <Rel Target="word_document.xml"/>
</Rels>
```

Or, in full:

```
<?xml version="1.0" encoding="utf-8"?>
<Relationships xmlns="[schema url]">
 <Relationship  Id="rId1"
                Target="word/document.xml"
                Type="[schema url]">
</Relationships>
```

It would seem that only the `word/document.xml` is essential, which makes sense. Even then, I was able further to remove some of the metadata from the tags ang get Word to open them, although it had to repair them. The most minimal file I could produce was this:

```
<Relationships>
 <Relationship  Target="word/document.xml"
                Type="[schema url]">
</Relationships>
```

The `<?xml?>` declaration, the `xmlns` attribute, and the `Id` attribute were all expendable, at the cost of having to open the file and have it be repaired.

Having done all this, I now question the utility of changing this file if it just has to be copied over anyway. Oh well.

1.5 Indices and tables

- genindex
- modindex
- search

d

f

p

Symbols

C

D

E

F

G

L

M

N

P

S

U

W