

# Text Layout Sample

This document demonstrates various styles that influence the way text is laid out and displayed.

## 1. Advanced Font-based Text Layout

By default PDFReactor does text layout for simple text, incl. Latin, in a performance optimized way that ignores some advanced font features. This can be changed for all text via the property `-ro-glyph-layout-mode`:

```
:root {
  -ro-glyph-layout-mode: quality;
}
```

This enables ligatures for fonts that support them. The following comparison shows the default glyph layout on top and quality glyph layout with ligatures on the bottom (the affected letter in all examples is the f):

prefix halfjacket reflex halfback wolfhound Kafka  
 prefix halfjacket reflex halfback wolfhound Kafka  
 buffer office cliffjumper waffle offbeat offhand fifififj  
 buffer office cliffjumper waffle offbeat offhand fifififj

## 2. Consistent Visual Font Size

When multiple fonts are used in the same line of text, either explicitly or because of font fallback, different sizes of letters may cause less than pleasing visual results and impact readability. Applying the style `font-size-adjust: -ro-from-font` to an element adjusts the fonts sizes of its children, so that they have the same x-height (lower-case letter size) as the font of the element itself.

Original:

Mixing fonts in a line can introduce an uneven appearance and decrease readability.

Mixing fonts in a line can introduce an uneven appearance and decrease readability.

With `"font-size-adjust: -ro-from-font"`:

Mixing fonts in a line can introduce an uneven appearance and decrease readability.

Mixing fonts in a line can introduce an uneven appearance and decrease readability.

### 3. Custom Line Breaking

By default line breaking is done according to Unicode rules, which cover many languages and writing systems. However, there are situations where modifying or overriding the breaking behavior will result in more pleasing output. In addition to common properties like `white-space` PDFreactor provides `-ro-line-break-opportunity`, allowing to adapt the behavior for the document or parts of it. The following examples illustrate different line breaking behaviors for a long URL (broken at every opportunity for demonstration purposes):

`http://host.tld/some/path-elements/file.ext?key1=val1&key2=val2`

Style	Result
<code>-ro-line-break-opportunity: normal</code> default equal to not specifying anything	<pre>http:// host.tld/ some/ path- elements/ file.ext? key1=val1&amp;key2=val2</pre>
<code>-ro-line-break-opportunity: normal '&amp;' / '-'</code> tweak breaking break after "&", but not after "-"	<pre>http:// host.tld/ some/ path-elements/ file.ext? key1=val1&amp; key2=val2</pre>
<code>-ro-line-break-opportunity: '' '[/?&amp;]'</code> replace breaking break before "/", "?" and "&"	<pre>http: / /host.tld /some /path-elements /file.ext ?key1=val1 &amp;key2=val2</pre>
<code>-ro-line-break-opportunity: '^[^/]' '[/?&amp;]'</code> <code>-ro-line-break-opportunity: '' '[/?&amp;]' / '/'</code> more advanced replacement also avoid breaking inside "/"	<pre>http: //host.tld /some /path-elements /file.ext ?key1=val1 &amp;key2=val2</pre>