## Sections

WordprocessingML does not natively store the concept of pages, since it is based on paragraphs and runs (which are laid out on to pages by consumers of this content). However, although there is no concept of storing pages in the WordprocessingML [format](format.docx), it is often necessary to store information about a page or [group](group.docx) of pages in a document, in order to store information that is to be used to [format](format.docx) the pages on which a set of paragraphs will appear. In WordprocessingML, this information is stored via the use of sections.

In WordprocessingML, sections are groupings of paragraphs that have a specific set of properties used to define the pages on which the text will appear, as well as other section-level (applying to all paragraphs' appearance) properties.

[Example: Consider a document with four paragraphs of text that is to be printed on a page in landscape mode, followed by ten paragraphs of text that are to be printed in portrait mode. This requirement implies information about the page(s) used to lay out each grouping of text—the first four paragraphs could require one page, or ten.

Therefore, rather than try to cache knowledge of the number of pages and their properties (which is likely to become invalid if the [XML](XML.docx) is manipulated by a producer that does not understand page layout), this information is stored by breaking the document into two sections, as follows:

<w:[p](p.docx)>
 …
</w:[p](p.docx)>

<w:[p](p.docx)>
 …
</w:[p](p.docx)>

<w:[p](p.docx)>
 …
</w:[p](p.docx)>

<w:[p](p.docx)>
 <w:[sectPr](sectPr.docx)>
 …
 (section one properties go here)
 <w:[pgSz](pgSz.docx) ... w:orient="landscape" />
 …
 </w:[sectPr](sectPr.docx)>
 …
</w:[p](p.docx)>

…

<w:[p](p.docx)>
 <w:[sectPr](sectPr.docx)>
 …
 (section two properties go here)
 <w:[pgSz](pgSz.docx) ... w:orient="landscape" />
 …
 </w:[sectPr](sectPr.docx)>
 …
</w:[p](p.docx)>

end example]