Docx4j headers/footers

This document describes certain aspects of header/footer handling. The immediate focus is on doing the right thing in PDF output. In due course, this document may be expanded to cover other things (eg header/footer creation).

There is a class DocumentModel, which contains:

**private** List<SectionWrapper> sections;

SectionWrapper stores, for each section (ie sequence of paragraphs followed by a sectPr element):

* PageDimensions page;
* HeaderFooterPolicy headerFooterPolicy;

together with the original SectPr element.

The HeaderFooterPolicy object can/should be used to find out what types of headers/footers are present (eg first/odd/even).

For more on the Word ML header/footer model, see the excerpt from the spec in the appendix. Note also the material on pgMar.

## Word UI limitations

Word won't let you set up the following:

Section 1

First page header

Other pages

Section 2

Odd & even header

Creating header structures using docx4j which you can't create directly in the Word UI is likely to lead to unexpected behaviour in Word!

## XSL FO concepts

Suppose you create a docx with the following structure:

Section 1

First page header

Other pages

Section 2

default header

If you use docx4j to create PDF output, you'll get an fo:page-sequence for each section:

<fo:page-sequence initial-page-number="1" format="1" master-reference="s1">

<fo:page-sequence initial-page-number="1" format="1" master-reference="s2">

The @master-reference refers to a page-sequence-master, for example:

<page-sequence-master master-name="s1">

<repeatable-page-master-alternatives>

<conditional-page-master-reference master-reference="s1-firstpage" page-position="first"/>

<conditional-page-master-reference master-reference="s1-default"/>

<conditional-page-master-reference master-reference="s1-simple"/>

</repeatable-page-master-alternatives>

</page-sequence-master>

Notice the conditional-page-master-reference elements, which in turn refer to simple-page-master elements, for example

<simple-page-master margin-bottom="0.25in" margin-left="1in" margin-right="1in" margin-top="0.25in" **master-name="s1-firstpage"** page-height="297mm" page-width="210mm">

<region-body margin-bottom="19mm" margin-left="0mm" margin-right="0mm" margin-top="19mm"/>

<region-before extent="12mm" region-name="xsl-region-before-firstpage"/>

<region-after extent="12mm" region-name="xsl-region-after-firstpage"/>

</simple-page-master>

The region-before and region-after elements refer to content which can be found back in the fo:page-sequence which had the @master-reference

Notice that you can have @margin-top on both simple-page-master and region-body

These values add together to determine where the body starts

## Differences between Open XML and XSL FO layout

### Open XML

The header is placed in the top margin.

This distance from the edge of the page to the start of the header is:

sectPr/w:pgMar/w:header (in twips)

Default is 720 (1/2 inch)

If the value of top margin is non-negative, then the text will be placed at the greater of:

• The value of top margin

• The extent of the header text

I think "extent of the header text" means spacing to top of header (header margin) + height of header content.

A negative value for top margin indicates that the contents of the main document shall be measured from the top of the page extent regardless of the header for that document, and therefore shall overlap the header text.

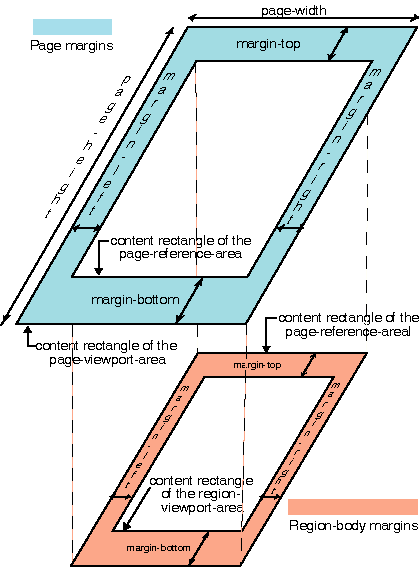
### XSL FO

**Region before & after live in region body margins:**

From <http://www.w3.org/TR/xsl/#fo_region-body>

The body region should be sized and positioned within the fo:simple-page-master so that there is room for the areas returned by the flow that is assigned to the fo:region-body and for any desired side regions, that is, fo:region-before, fo:region-after, fo:region-start and fo:region-end's that are to be placed on the same page. These side regions are positioned within the content-rectangle of the page-reference-area. The margins on the fo:region-body are used to position the region-viewport-area for the fo:region-body and to leave space for the other regions that surround the fo:region-body.

ie the region before goes in the region body margin



The spacing between the last four regions and the fo:region-body is determined by subtracting the relevant extent trait on the side regions from the trait that corresponds to the "margin-x" property on the fo:region-body.

## LayoutMasterSetBuilder

Docx4j contains a class org.docx4j.convert.out.pdf.viaXSLFO. LayoutMasterSetBuilder. It is this class which is responsible for generating the LayoutMasterSet (which contains page-sequence-master elements etc).

It needs to handle the three distinct cases described by the WordML spec.

Case 1: top margin non-negative, and extent of the header text greater than value of top margin

Done

Case 2: top margin non-negative, and extent of the header text less than value of top margin

Done

Case 3: negative value for top margin

TODO

For each of these cases, we need appropriate XSL FO.

Our approach (post v3.0.1, implemented March 2014) is as follows

\* Its not possible to let FOP set the height (@extent) of the header and footer regions

\* automatically: http://apache-fop.1065347.n5.nabble.com/Auto-size-header-to-fit-with-content-td4455.html

\*

\* So we need to set the height (@extent) of the header and footer regions explicitly.

\*

\* We do that by creating a temp FO file which contains essentially just the headers/footers,

\* then interrogating FOP's area tree representation to find the height of each.

\*

\* So:

\* 1. create LayoutMasterSet (with large extents)

See LayoutMasterSetBuilder. createSimplePageMaster

\* 2. generate area tree from that

See LayoutMasterSetBuilder. getLayoutMasterSetFragment and its calls to FOPAreaTreeHelper

## \* 3. return LayoutMasterSet (with required extents)

See FOPAreaTreeHelper. adjustLayoutMasterSet

Note that where there is a header or footer, we set:

* simple page master margin top to Word’s space between page edge and header
* the region before’s extent to be what it needs to be, and the region body top margin to fit that
* if Word’s top margin is bigger than the above, we make the region body top margin bigger to match

## Test cases

To test docx4j's handling of headers/footers in PDF output, we should have the following test cases:

1. simple\_hf\_none: A simple docx (one section) with no headers/footers
2. simple\_hf: A simple docx (one section)with a simple header/footer (ie less than the value of top margin)
3. simple\_hf\_long: A simple docx (one section)with a header containing many lines of content (eg 12 say)
4. simple\_hf\_margin: A simple docx (one section)with a simple header/footer and a header pgMar setting
5. simple\_hf\_negative: A simple docx (one section)with a simple header/footer and a negative value for top margin
6. different\_first\_page: A docx with a different first page header
7. odd\_even: A docx with different odd/even headers
8. sections\_multiple: A docx with multiple sections
9. odd\_even\_different\_first\_page: A docx with a different first page header

Appendix: Excerpt from OpenXML spec

## Headers and Footers

Headers and footers refer to text, graphics or data (such as page number, date, document title, and so on) that can appear at the top or bottom of each page in a WordprocessingML document.

A header appears in the top margin (above the main document content on the page), while a footer appears in the bottom margin of a document page (below the main document content on the page).



Since WordprocessingML is a flow-based format, headers and footers are applied by specifying the headers and footers for all pages in a particular section of a document.

Within each section of a document there may be up to three different types of headers and footers:

* First page header/footer
* Odd page header/footer
* Even page header/footer

First page headers and footers specify a unique header or footer which shall appear on the first page of a section. Odd page headers and footers specify a unique header and footer which shall appear on all odd numbered pages for a given section. Even page headers and footers specify a unique header and footer which shall appear on all even numbered pages in a given section.

### evenAndOddHeaders (Different Even/Odd Page Headers and Footers)

This element specifies whether sections in this document shall have different headers and footers for even and odd pages (an odd page header/footer and an even page header/footer).

If the val attribute is set to true, then each section in the document shall use an odd page header for all odd numbered pages in the section, and an even page header for all even numbered pages in the section (counting each page in the section starting from one, regardless of the page numbering settings for the section). If the val attribute is set to false, then all pages in a section shall use the odd page header.

This setting does not affect the presence of a first page header on each section, which is specified using the titlePg element (§0). If a first page header is specified, then all subsequent pages shall have this setting applied, including the first page in the odd/even page count.

If this element is set to false and an even page header is specified , then it shall be ignored and only the odd page header shall be displayed. Conversely, if this element is set to true and either header type is omitted for a given section, then a blank header shall be created as needed (another header type shall not be used in its place).

If this element is omitted, then its value shall be assumed to be false.

[Example: Consider a document which shall have a different even and odd page header for each section in its contents. This requirement must be specified using the following WordprocessingML:

<w:settings>  
 …  
 <w:evenAndOddHeaders />  
 …  
</w:setttings>

Since the evenAndOddHeaders property is set (and its default value is true), this document will now have different headers and footers for even and odd pages. end example]

|  |
| --- |
| Parent Elements |
| settings (§**Error! Reference source not found.**) |

|  |  |
| --- | --- |
| Attributes | Description |
| val (On/Off Value) | Specifies a binary value for the property defined by the parent XML element.  A value of on, 1, or true specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted.  A value of off, 0, or false specifies that the property shall be explicitly turned off.  [Example: For example, consider the following on/off property:  <w:… w:val="off"/>  The val attribute explicitly declares that the property is turned off. end example]  The possible values for this attribute are defined by the ST\_OnOff simple type (§**Error! Reference source not found.**). |

The following XML Schema fragment defines the contents of this element:

<complexType name="CT\_OnOff">

<attribute name="val" type="ST\_OnOff"/>

</complexType>

### footerReference (Footer Reference)

This element specifies a single footer which shall be associated with the current section in the document. This footer shall be referenced via the id attribute, which specifies an explicit relationship to the appropriate Footer part in the WordprocessingML package.

If the relationship type of the relationship specified by this element is not http://schemas.openxmlformats.org/officeDocument/2006/footer, is not present, or does not have a TargetMode attribute value of Internal, then the document shall be considered non-conformant.

Within each section of a document there may be up to three different types of footers:

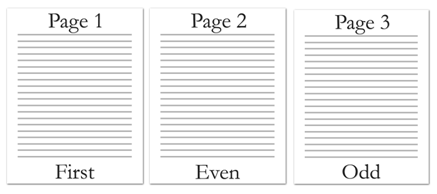
* First page footer
* Odd page footer
* Even page footer

The footer type specified by the current footerReference is specified via the type attribute.

If any type of footer is omitted for a given section, then the following rules shall apply.

* If no footerReference for the first page footer is specified and the titlePg element is specified, then the first page footer shall be inherited from the previous section or, if this is the first section in the document, a new blank footer shall be created. If the titlePg element is not specified, then no first page footer shall be shown, and the odd page footer shall be used in its place.
* If no footerReference for the even page footer is specified and the evenAndOddHeaders element is specified, then the even page footer shall be inherited from the previous section or, if this is the first section in the document, a new blank footer shall be created. If the evenAndOddHeaders element is not specified, then no even page footer shall be shown. and the odd page footer shall be used in its place.
* If no footerReference for the odd page footer is specified then the even page footer shall be inherited from the previous section or, if this is the first section in the document, a new blank footer shall be created.

[Example: Consider a three page document with different first, odd, and even page footers defined as follows:



This document defines three footers, each of have a relationship from the document part with a unique relationship ID, as shown in the following packaging markup:

<Relationships xmlns=http://schemas.openxmlformats.org/package/2006/relationships>

…

<Relationship Id="rId6" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer" Target="footer1.xml" />

<Relationship Id="rId7" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer" Target="footer2.xml" />

<Relationship Id="rId10" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer" Target="footer3.xml" />

…

</Relationships>

These relationships are then referenced in the section's properties using the following WordprocessingML:

<w:sectPr>

…

<w:footerReference r:id="rId6" w:type="first" />

<w:footerReference r:id="rId7" w:type="default" />

<w:footerReference r:id="rId10" w:type="even" />

…

</w:sectPr>

The resulting section shall use the footer part with relationship id rId6 for the first page, the footer part with relationship id rId10 for all subsequent even pages, and the footer part with relationship id rId7 for all subsequent odd pages. end example]

|  |
| --- |
| Parent Elements |
| sectPr (§**Error! Reference source not found.**); sectPr (§**Error! Reference source not found.**) |

|  |  |
| --- | --- |
| Attributes | Description |
| id (Relationship to Part)  Namespace: .../officeDocument/2006/relationships | Specifies the relationship ID to a specified part.  The specified relationship shall match the type required by the parent element:   * http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer for the footerReference element * http://schemas.openxmlformats.org/officeDocument/2006/relationships/header for the headerReference element * http://schemas.openxmlformats.org/officeDocument/2006/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements * http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings for the printerSettings element   [Example: Consider an XML element which has the following id attribute:  <… r:id="rId10" />  The markup specifies the associated relationship part with relationship ID rId1 contains the corresponding relationship information for the parent XML element. end example]  The possible values for this attribute are defined by the ST\_RelationshipId simple type (§**Error! Reference source not found.**). |
| type (Header or Footer Type) | Specifies the type of header or footer specified by the target relationship ID. This type determines the page(s) on which the current header or footer shall be displayed.  If any section contains more than a single header or footer of each type, then the document shall be considered non-conformant.  [Example: Consider a document with the following WordprocessingML:  <w:sectPr>  …  <w:footerReference r:id="rId6" w:type="first" />  <w:footerReference r:id="rId7" w:type="first" />  <w:footerReference r:id="rId10" w:type="even" />  …  </w:sectPr>  The resulting section has two footers of type first, and therefore is invalid. end example]  [Example: Consider a WordprocessingML section which specifies the following header reference:  <w:headerReference r:id="rId10" w:type="first" />  The resulting section shall use the specified header part for the first page. end example]  The possible values for this attribute are defined by the ST\_HdrFtr simple type (§**Error! Reference source not found.**). |

The following XML Schema fragment defines the contents of this element:

<complexType name="CT\_HdrFtrRef">

<complexContent>

<extension base="CT\_Rel">

<attribute name="type" type="ST\_HdrFtr" use="required"/>

</extension>

</complexContent>

</complexType>

### ftr (Footer)

This element specifies the content for a single footer for use within one or more sections of a WordprocessingML document.

Within the ftr element, the content of the element is similar to the content of the body (§**Error! Reference source not found.**) element, and contains what is referred to as block-level markup - markup which can exist as a sibling element to paragraphs in a WordprocessingML document.

[Example: Consider the following simple one page document with one footer:



This document defines one footer with the text footer. The footer contents are stored in a unique footer part. The resulting footer is represented by the following WordprocessingML:

<w:ftr>  
 <w:p>  
 <w:r>  
 <w:t>Footer</w:t>  
 </w:r>  
 </w:p>  
</w:ftr>

Since footers are containers of block level contents, all block level elements can be used within them. In this particular example, the content is a single paragraph. end example]

[Example: Consider a more complex three page document with different first, odd, and even page footers defined:



This document defines three footers stored in three different footer parts. The resulting footers are represented by the following WordprocessingML:

First page footer part:

<w:ftr>  
 <w:p>  
 <w:r>  
 <w:t>First</w:t>  
 </w:r>  
 </w:p>  
</w:ftr>

Even page footer part:

<w:ftr>  
 <w:p>  
 <w:r>  
 <w:t>Even</w:t>  
 </w:r>  
 </w:p>  
</w:ftr>

Odd page footer part:

<w:ftr>  
 <w:p>  
 <w:r>  
 <w:t>Odd</w:t>  
 </w:r>  
 </w:p>  
</w:ftr>

end example]

|  |
| --- |
| Parent Elements |
| Root element of WordprocessingML Footer part |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| altChunk (Anchor for Imported External Content) | §**Error! Reference source not found.** |
| bookmarkEnd (Bookmark End) | §**Error! Reference source not found.** |
| bookmarkStart (Bookmark Start) | §**Error! Reference source not found.** |
| commentRangeEnd (Comment Anchor Range End) | §**Error! Reference source not found.** |
| commentRangeStart (Comment Anchor Range Start) | §**Error! Reference source not found.** |
| customXml (Block-Level Custom XML Element) | §**Error! Reference source not found.** |
| customXmlDelRangeEnd (Custom XML Markup Deletion End) | §**Error! Reference source not found.** |
| customXmlDelRangeStart (Custom XML Markup Deletion Start) | §**Error! Reference source not found.** |
| customXmlInsRangeEnd (Custom XML Markup Insertion End) | §**Error! Reference source not found.** |
| customXmlInsRangeStart (Custom XML Markup Insertion Start) | §**Error! Reference source not found.** |
| customXmlMoveFromRangeEnd (Custom XML Markup Move Source End) | §**Error! Reference source not found.** |
| customXmlMoveFromRangeStart (Custom XML Markup Move Source Start) | §**Error! Reference source not found.** |
| customXmlMoveToRangeEnd (Custom XML Markup Move Destination Location End) | §**Error! Reference source not found.** |
| customXmlMoveToRangeStart (Custom XML Markup Move Destination Location Start) | §**Error! Reference source not found.** |
| del (Deleted Run Content) | §**Error! Reference source not found.** |
| ins (Inserted Run Content) | §**Error! Reference source not found.** |
| moveFrom (Move Source Run Content) | §**Error! Reference source not found.** |
| moveFromRangeEnd (Move Source Location Container - End) | §**Error! Reference source not found.** |
| moveFromRangeStart (Move Source Location Container - Start) | §**Error! Reference source not found.** |
| moveTo (Move Destination Run Content) | §**Error! Reference source not found.** |
| moveToRangeEnd (Move Destination Location Container - End) | §**Error! Reference source not found.** |
| moveToRangeStart (Move Destination Location Container - Start) | §**Error! Reference source not found.** |
| oMath (Office Math) | §**Error! Reference source not found.** |
| oMathPara (Math Paragraph) | §**Error! Reference source not found.** |
| p (Paragraph) | §**Error! Reference source not found.** |
| permEnd (Range Permission End) | §**Error! Reference source not found.** |
| permStart (Range Permission Start) | §**Error! Reference source not found.** |
| proofErr (Proofing Error Anchor) | §**Error! Reference source not found.** |
| sdt (Block-Level Structured Document Tag) | §**Error! Reference source not found.** |
| tbl (Table) | §**Error! Reference source not found.** |

The following XML Schema fragment defines the contents of this element:

<complexType name="CT\_HdrFtr">

<group ref="EG\_BlockLevelElts" minOccurs="1" maxOccurs="unbounded"/>

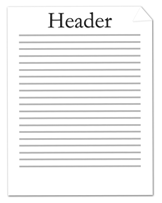
</complexType>

### hdr (Header)

This element specifies the content for a single header for use within one or more sections of a WordprocessingML document.

Within the hdr element, the content of the element is similar to the content of the body (§**Error! Reference source not found.**) element, and contains what is referred to as block-level markup - markup which can exist as a sibling element to paragraphs in a WordprocessingML document.

[Example: Consider the following simple one page document with one header:

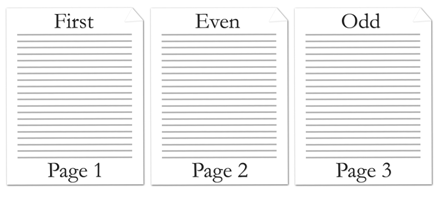


This document defines one header with the text Header. The header's contents is stored in a unique Header part. The resulting header is represented by the following WordprocessingML:

<w:hdr>  
 <w:p>  
 <w:r>  
 <w:t>Header</w:t>  
 </w:r>  
 </w:p>  
</w:hdr>

Since headers are containers of block level contents, all block level elements can be used within them. In this particular example, the content is a single paragraph. end example]

[Example: Consider a more complex three page document with different first, odd, and even page headers defined:



This document defines three headers stored in three different header parts. The resulting headers are represented by the following WordprocessingML:

First page header part:

<w:hdr>  
 <w:p>  
 <w:r>  
 <w:t>First</w:t>  
 </w:r>  
 </w:p>  
</w:hdr>

Even page header part:

<w:hdr>  
 <w:p>  
 <w:r>  
 <w:t>Even</w:t>  
 </w:r>  
 </w:p>  
</w:hdr>

Odd page header part:

<w:hdr>  
 <w:p>  
 <w:r>  
 <w:t>Odd</w:t>  
 </w:r>  
 </w:p>  
</w:hdr>

end example]

|  |
| --- |
| Parent Elements |
| Root element of WordprocessingML Header part |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| altChunk (Anchor for Imported External Content) | §**Error! Reference source not found.** |
| bookmarkEnd (Bookmark End) | §**Error! Reference source not found.** |
| bookmarkStart (Bookmark Start) | §**Error! Reference source not found.** |
| commentRangeEnd (Comment Anchor Range End) | §**Error! Reference source not found.** |
| commentRangeStart (Comment Anchor Range Start) | §**Error! Reference source not found.** |
| customXml (Block-Level Custom XML Element) | §**Error! Reference source not found.** |
| customXmlDelRangeEnd (Custom XML Markup Deletion End) | §**Error! Reference source not found.** |
| customXmlDelRangeStart (Custom XML Markup Deletion Start) | §**Error! Reference source not found.** |
| customXmlInsRangeEnd (Custom XML Markup Insertion End) | §**Error! Reference source not found.** |
| customXmlInsRangeStart (Custom XML Markup Insertion Start) | §**Error! Reference source not found.** |
| customXmlMoveFromRangeEnd (Custom XML Markup Move Source End) | §**Error! Reference source not found.** |
| customXmlMoveFromRangeStart (Custom XML Markup Move Source Start) | §**Error! Reference source not found.** |
| customXmlMoveToRangeEnd (Custom XML Markup Move Destination Location End) | §**Error! Reference source not found.** |
| customXmlMoveToRangeStart (Custom XML Markup Move Destination Location Start) | §**Error! Reference source not found.** |
| del (Deleted Run Content) | §**Error! Reference source not found.** |
| ins (Inserted Run Content) | §**Error! Reference source not found.** |
| moveFrom (Move Source Run Content) | §**Error! Reference source not found.** |
| moveFromRangeEnd (Move Source Location Container - End) | §**Error! Reference source not found.** |
| moveFromRangeStart (Move Source Location Container - Start) | §**Error! Reference source not found.** |
| moveTo (Move Destination Run Content) | §**Error! Reference source not found.** |
| moveToRangeEnd (Move Destination Location Container - End) | §**Error! Reference source not found.** |
| moveToRangeStart (Move Destination Location Container - Start) | §**Error! Reference source not found.** |
| oMath (Office Math) | §**Error! Reference source not found.** |
| oMathPara (Math Paragraph) | §**Error! Reference source not found.** |
| p (Paragraph) | §**Error! Reference source not found.** |
| permEnd (Range Permission End) | §**Error! Reference source not found.** |
| permStart (Range Permission Start) | §**Error! Reference source not found.** |
| proofErr (Proofing Error Anchor) | §**Error! Reference source not found.** |
| sdt (Block-Level Structured Document Tag) | §**Error! Reference source not found.** |
| tbl (Table) | §**Error! Reference source not found.** |

The following XML Schema fragment defines the contents of this element:

<complexType name="CT\_HdrFtr">

<group ref="EG\_BlockLevelElts" minOccurs="1" maxOccurs="unbounded"/>

</complexType>

### headerReference (Header Reference)

This element specifies a single header which shall be associated with the current section in the document. This header shall be referenced via the id attribute, which specifies an explicit relationship to the appropriate Header part in the WordprocessingML package.

If the relationship type of the relationship specified by this element is not http://schemas.openxmlformats.org/officeDocument/2006/header, is not present, or does not have a TargetMode attribute value of Internal, then the document shall be considered non-conformant.

Within each section of a document there may be up to three different types of headers:

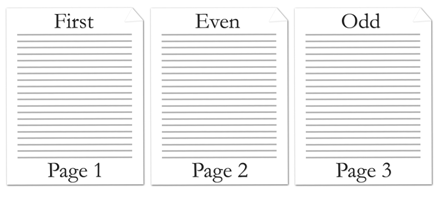
* First page header
* Odd page header
* Even page header

The header type specified by the current headerReference is specified via the type attribute.

If any type of header is omitted for a given section, then the following rules shall apply.

* If no headerReference for the first page header is specified and the titlePg element is specified, then the first page header shall be inherited from the previous section or, if this is the first section in the document, a new blank header shall be created. If the titlePg element is not specified, then no first page header shall be shown, and the odd page header shall be used in its place.
* If no headerReference for the even page header is specified and the evenAndOddHeaders element is specified, then the even page header shall be inherited from the previous section or, if this is the first section in the document, a new blank header shall be created. If the evenAndOddHeaders element is not specified, then no even page header shall be shown, and the odd page header shall be used in its place.
* If no headerReference for the odd page header is specified then the even page header shall be inherited from the previous section or, if this is the first section in the document, a new blank header shall be created.

[Example: Consider a three page document with different first, odd, and even page header defined as follows:



This document defines three headers, each of have a relationship from the document part with a unique relationship ID, as shown in the following packaging markup:

<Relationships xmlns=http://schemas.openxmlformats.org/package/2006/relationships>

…

<Relationship Id="rId2" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/header" Target="header1.xml" />

<Relationship Id="rId3" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/header" Target="header2.xml" />

<Relationship Id="rId5" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/header" Target="header3.xml" />

…

</Relationships>

These relationships are then referenced in the section's properties using the following WordprocessingML:

<w:sectPr>

…

<w:headerReference r:id="rId3" w:type="first" />

<w:headerReference r:id="rId5" w:type="default" />

<w:headerReference r:id="rId2" w:type="even" />

…

</w:sectPr>

The resulting section shall use the header part with relationship id rId3 for the first page, the header part with relationship id rId2 for all subsequent even pages, and the header part with relationship id rId5 for all subsequent odd pages. end example]

|  |
| --- |
| Parent Elements |
| sectPr (§**Error! Reference source not found.**); sectPr (§**Error! Reference source not found.**) |

|  |  |
| --- | --- |
| Attributes | Description |
| id (Relationship to Part)  Namespace: .../officeDocument/2006/relationships | Specifies the relationship ID to a specified part.  The specified relationship shall match the type required by the parent element:   * http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer for the footerReference element * http://schemas.openxmlformats.org/officeDocument/2006/relationships/header for the headerReference element * http://schemas.openxmlformats.org/officeDocument/2006/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements * http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings for the printerSettings element   [Example: Consider an XML element which has the following id attribute:  <… r:id="rId10" />  The markup specifies the associated relationship part with relationship ID rId1 contains the corresponding relationship information for the parent XML element. end example]  The possible values for this attribute are defined by the ST\_RelationshipId simple type (§**Error! Reference source not found.**). |
| type (Header or Footer Type) | Specifies the type of header or footer specified by the target relationship ID. This type determines the page(s) on which the current header or footer shall be displayed.  If any section contains more than a single header or footer of each type, then the document shall be considered non-conformant.  [Example: Consider a document with the following WordprocessingML:  <w:sectPr>  …  <w:footerReference r:id="rId6" w:type="first" />  <w:footerReference r:id="rId7" w:type="first" />  <w:footerReference r:id="rId10" w:type="even" />  …  </w:sectPr>  The resulting section has two footers of type first, and therefore is invalid. end example]  [Example: Consider a WordprocessingML section which specifies the following header reference:  <w:headerReference r:id="rId10" w:type="first" />  The resulting section shall use the specified header part for the first page. end example]  The possible values for this attribute are defined by the ST\_HdrFtr simple type (§**Error! Reference source not found.**). |

The following XML Schema fragment defines the contents of this element:

<complexType name="CT\_HdrFtrRef">

<complexContent>

<extension base="CT\_Rel">

<attribute name="type" type="ST\_HdrFtr" use="required"/>

</extension>

</complexContent>

</complexType>

### titlePg (Different First Page Headers and Footers)

This element specifies whether the parent section in this document shall have a different header and footer for its first page.

If the val attribute is set to true, then the parent section in the document shall use a first page header for the first page in the section. If the val attribute is set to false, then the first page in the parent section shall use the odd page header.

This setting does not affect the presence of even and odd page header on all sections, which is specified using the evenAndOddHeaders element (§0).

If this element is set to false and a first page header is specified , then it shall be ignored and only the odd page header shall be displayed. Conversely, if this element is set to true and the first page header type is omitted for the given section, then a blank header shall be created as needed (another header type shall not be used in its place).

If this element is omitted, then its value shall be assumed to be false.

[Example: Consider a section which shall have a different first page header. This requirement is specified using the following WordprocessingML:

<w:sectPr>  
 …  
 <w:titlePg />  
 …  
</w:sectPr>

Since the titlePg property is present (and its default attribute value is true), this document will now have a different header and footer for its first page. end example]

|  |
| --- |
| Parent Elements |
| sectPr (§**Error! Reference source not found.**); sectPr (§**Error! Reference source not found.**); sectPr (§**Error! Reference source not found.**) |

|  |  |
| --- | --- |
| Attributes | Description |
| val (On/Off Value) | Specifies a binary value for the property defined by the parent XML element.  A value of on, 1, or true specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted.  A value of off, 0, or false specifies that the property shall be explicitly turned off.  [Example: For example, consider the following on/off property:  <w:… w:val="off"/>  The val attribute explicitly declares that the property is turned off. end example]  The possible values for this attribute are defined by the ST\_OnOff simple type (§**Error! Reference source not found.**). |

The following XML Schema fragment defines the contents of this element:

<complexType name="CT\_OnOff">

<attribute name="val" type="ST\_OnOff"/>

</complexType>

### pgMar (Page Margins)

This element specifies the page margins for all pages in this section.

[Example: Consider a page with a one-inch margin on all sides. Specifying these margins requires the following WordprocessingML:

<w:sectPr>  
 <w:pgMar w:bottom="1440" w:top="1440" w:right="1440" w:left="1440"/>  
 …  
</w:sectPr>

This section specifies page margins of 1,440 twentieths of a point (one inch) on all sides. end example]

|  |
| --- |
| Parent Elements |
| sectPr (§**Error! Reference source not found.**); sectPr (§**Error! Reference source not found.**); sectPr (§**Error! Reference source not found.**) |

|  |  |
| --- | --- |
| Attributes | Description |
| bottom (Page Bottom Spacing) | Specifies the distance (in twentieths of a point) between the bottom of the text margins for the main document and the bottom of the page for all pages in this section.  If the value of bottom is non-negative, then the text will be placed at the greater of:   * The value of bottom * The extent of the footer text   [Example: Consider a document where the footer shall start one inch of the bottom of the page extent, but the contents of the main document story specify that they shall start one-half of an inchh from the page extents. To specify these boundaries, the following page margins may specified in the WordprocessingML:  <w:pgMar ... w:bottom="720"  w:footer="1440"/>  This fragment specifies that the footer shall start 1440 twentieths of a point from the bottom of the page, but the main document story shall start 720 twentieths of a point from the bottom of the page. Since the footer extent is guaranteed to be greater in this case, the bottom text extent ends at the top of the footer region. end example]  A negative value indicates that the contents of the main document shall be measured from the bottom of the page extent regardless of the footer for that document, and therefore shall overlap the footer text.  [Example: Consider a document where the footer shall start one inch of the bottom of the page extent, but the contents of the main document story shall start one-half of an inch from the page extents. To specify these boundaries, the following page margins may specified in the WordprocessingML:  <w:pgMar ... w:bottom="-720"  w:footer="1440"/>  This fragment specifies that the footer shall start 1440 twentieths of a point from the bottom of the page, and the main document story shall start 720 twentieths of a point from the bottom of the page. Since the value of bottom is negative in this case, the bottom text extent starts one-half of an inch from the bottom of the page and overlaps any footer text. end example]  The possible values for this attribute are defined by the ST\_SignedTwipsMeasure simple type (§**Error! Reference source not found.**). |
| footer (Spacing to Bottom of Footer) | Specifies the distance (in twentieths of a point) from the bottom edge of the page to the bottom edge of the footer.  [Example: Consider a document where the footer shall start one inch above the bottom of the page extent.  To specify this boundary, the following page margins must specified in the WordprocessingML:  <w:pgMar … w:footer="1440"/>  This fragment specifies that the footer shall start 1440 twentieths of a point from the bottom of the page. end example]  The possible values for this attribute are defined by the ST\_TwipsMeasure simple type (§**Error! Reference source not found.**). |
| gutter (Page Gutter Spacing) | Specifies the page gutter for each page in the current section.  The page gutter defines the amount of extra space added to the specified margin, above any existing margin values. [Note: This setting is typically used when a document is being created for binding, in order to ensure that the resulting margins are present after the binding gutter is consumed by the printed matter binding. end note]  [Example: Consider a document where the margin shall start one inch of the left edge of the page extent after one-half of an inch is hidden by the page binding.  To specify this condition, a user could simply use a left margin of 1.5 inches, which would be lost if the margins are later changed, or could specify a one-half inch gutters follows in the WordprocessingML:  <w:pgMar … w:gutter="720"/>  This fragment specifies that the gutter shall span 720 twentieths of a point, after which any margin value shall be added. end example]  The possible values for this attribute are defined by the ST\_TwipsMeasure simple type (§**Error! Reference source not found.**). |
| header (Spacing to Top of Header) | Specifies the distance (in twentieths of a point) from the top edge of the page to the top edge of the header.  [Example: Consider a document where the header shall start two inches below the top of the page extent. To specify this boundary, the following page margins must specified in the WordprocessingML:  <w:pgMar … w:header="2880"/>  This fragment specifies that the header shall start 2880 twentieths of a point from the top of the page. end example]  The possible values for this attribute are defined by the ST\_TwipsMeasure simple type (§**Error! Reference source not found.**). |
| left (Left Margin Spacing) | Specifies the distance (in twentieths of a point) between the left edge of the page and the left edge of the text extents for this document.  [Example: Consider a document where the left text extent shall start two inches inside the page extent. To specify this boundary, the following page margins must specified in the WordprocessingML:  <w:pgMar … w:left="2880"/>  This fragment specifies that the left margin shall span 2880 twentieths of a point from the left edge of the page. end example]  The possible values for this attribute are defined by the ST\_TwipsMeasure simple type (§**Error! Reference source not found.**). |
| right (Right Margin Spacing) | Specifies the distance (in twentieths of a point) between the right edge of the page and the right edge of the text extents for this document.  [Example: Consider a document where the right text extent shall start one inch inside the page.  To specify this boundary, the following page margins must specified in the WordprocessingML:  <w:pgMar … w:right="1440"/>  This fragment specifies that the right margin shall span 1440 twentieths of a point from the right edge of the page. end example]  The possible values for this attribute are defined by the ST\_TwipsMeasure simple type (§**Error! Reference source not found.**). |
| top (Top Margin Spacing) | Specifies the distance (in twentieths of a point) between the top of the text margins for the main document and the top of the page for all pages in this section.  If the value of top is non-negative, then the text will be placed at the greater of:   * The value of top * The extent of the header text   [Example: Consider a document where the header shall start one inch from the top of the page extent, but the contents of the main document story specify that they shall start one-half of an inch from the page extents. To specify these boundaries, the following page margins may specified in the WordprocessingML:  <w:pgMar … w:top="720" w:header="1440"/>  This fragment specifies that the header shall start 1440 twentieths of a point from the top of the page, but the main document story shall start 720 twentieths of a point from the top of the page. Since the header extent is guaranteed to be greater in this case, the main text extent ends at the bottom of the header region. end example]  A negative value indicates that the contents of the main document shall be measured from the top of the page extent regardless of the header for that document, and therefore shall overlap the header text.  [Example: Consider a document where the header shall start one inch from the top of the page extent, but the contents of the main document story shall start one-half of an inch from the page extents. To specify these boundaries, the following page margins may specified in the WordprocessingML:  <w:pgMar ... w:top="-720"  w:header="1440"/>  This fragment specifies that the header shall start 1440 twentieths of a point from the top of the page, and the main document story shall start 720 twentieths of a point from the top of the page. Since the value of top is negative in this case, the top text extent starts one-half of an inch from the top of the page and overlaps any header text. end example]  The possible values for this attribute are defined by the ST\_SignedTwipsMeasure simple type (§**Error! Reference source not found.**). |

The following XML Schema fragment defines the contents of this element:

<complexType name="CT\_PageMar">

<attribute name="top" type="ST\_SignedTwipsMeasure" use="required"/>

<attribute name="right" type="ST\_TwipsMeasure" use="required"/>

<attribute name="bottom" type="ST\_SignedTwipsMeasure" use="required"/>

<attribute name="left" type="ST\_TwipsMeasure" use="required"/>

<attribute name="header" type="ST\_TwipsMeasure" use="required"/>

<attribute name="footer" type="ST\_TwipsMeasure" use="required"/>

<attribute name="gutter" type="ST\_TwipsMeasure" use="required"/>

</complexType>