

Section

Description

This class is used to represent a section of the document. Many documents will only contain one section. More complex documents will divide the document into multiple sections. A section can be roughly correlated to a chapter in a book. Each section can have different formatting settings, including:

C#

```
public sealed class Section : Element
```

vb.net

```
Public NotInheritable Class Section  
    Inherits Element
```

Remarks

To create a new section, use [Element.CreateSectionBefore\(\)](#).

To get an existing section, use [Document.Sections](#).

The following sample shows you how to get the add a new section to a document and how to get the first section of a document.

Examples

C#

```
//--- Add a new section to a document  
WordApplication app = new WordApplication();  
Document doc = app.Open(@"C:\sample.doc");  
Section sect = doc.CreateSectionAfter();  
  
//--- Get the first section of a document  
WordApplication app = new WordApplication();  
Document doc = app.Open(@"C:\sample.doc");  
Section sect1 = doc.Sections[0];
```

vb.net

```
'--- Add a new section to a document
Dim app As New WordApplication()
Dim doc As Document = app.Open("C:\sample.doc")
Dim sect As Section = doc.CreateSectionAfter()

'--- Get the first section of a document
Dim app As New WordApplication()
Dim doc As Document = app.Open("C:\sample.doc")
Dim sect As Section = doc.Sections(0)
```

Properties

Name	Description
BorderAlwaysInFront	Sets or returns a <code>boolean</code> that represents whether or not page borders for this section will always display in front of any objects it may overlap.
BorderAppliedTo	Sets or returns a BorderArea object that represents where the page border will be applied to.
BorderOffsetsFromPage	Sets or returns a <code>boolean</code> that represents whether or not the padding for this section's page border will be measured from the edge of the page or from the text it surrounds.
BottomMargin	Sets or returns an <code>int</code> that represents the vertical distance of the bottom margin from the bottom edge of the page for this section. One twip = (1/20 pt) or (1/1440 in)
Break	Sets or returns a Section.BreakType object that represents the type of break (before) this section. There are 5 types of breaks: Continuous, Column, Page, EvenPage, and OddPage.
FirstLineNum	Sets or returns an <code>int</code> that represents the number that the line numbering starts at for this section.
FirstPageNum	Sets or returns an <code>int</code> that represents the number that the page numbering starts at for this section.
FooterY	Sets or returns an <code>int</code> that represents the y coordinate of the footer measured from the top edge of the page. One twip = (1/20 pt) or (1/1440 in)
GutterWidth	Sets or returns an <code>int</code> that represents the gutter width for this section. The gutter adds space to the margin. Gutters are used when a document needs additional space to allow for binding. One twip = (1/20 pt) or (1/1440 in)
HasTitlePage	Sets or returns a <code>boolean</code> that represents whether or not this section has a title page. This can affect page numbering and the appearance of headers and footers.
HeaderY	Sets or returns an <code>int</code> that represents the y coordinate of the header measured from the top edge of the page. One twip = (1/20 pt) or (1/1440 in)
LeftMargin	Sets or returns an <code>int</code> that represents the horizontal distance of the left margin from the left edge of the page for this section. One twip = (1/20 pt) or (1/1440 in)

LineNumberingType	Sets or returns a Section.LineNumbering object that represents the line numbering restart policy for this section. There are 3 options:
LineNumberX	Sets or returns an <code>int</code> that represents the x position of line numbers in twips. One twip = (1/20 pt) or (1/1440 in)
LineNumModulus	Sets or returns an <code>int</code> that represents the line numbering modulus. This setting is used to set whether line numbers display on every line, every 2nd line, etc. Setting this to 0 (zero) will display no line numbers. Setting this to 1 will display a number by every line. Setting this to 2 will display a number by every 2nd line, etc.
LinesBetween	Sets or returns a <code>boolean</code> that represents whether or not Word will draw vertical lines between columns.
NumColumns	Sets or returns an <code>int</code> that represents the number of columns in this section.
PageHeight	Sets or returns an <code>int</code> that represents the page height for this section in twips. One twip = (1/20 pt) or (1/1440 in)
PageNumberFormat	Sets or returns a Section.NumberFormat object that represents the page number format for this section. There are 5 type of page number formats:
PageOrientation	Sets or returns a Section.Orientation object that represents the page orientation for this section. Possible values: Portrait, Landscape, Mixed
PageSize	Sets or returns the size of printed pages.
PageWidth	Sets or returns an <code>int</code> that represents the page width for this section in twips. One twip = (1/20 pt) or (1/1440 in)
RestartPageNum	Sets or returns a <code>boolean</code> that represents whether or not page numbering will restart at the beginning of this section.
RightMargin	Sets or returns an <code>int</code> that represents the horizontal distance of the right margin from the right edge of the page for this section. One twip = (1/20 pt) or (1/1440 in)
TopMargin	Sets or returns an <code>int</code> that represents the vertical distance of the top margin from the top edge of the page for this section. One twip = (1/20 pt) or (1/1440 in)
Unlocked	Sets or returns a <code>boolean</code> that represents whether or not this section is unlocked.

Methods

Name	Description
GetBorder(Border.Location)	Returns a Border object that represents the border of this section at the specified location. You can use the returned object to manipulate the borders for this section. Valid locations: Top, Left, Bottom, Right
GetFooter(Section.HeaderFooterType)	Returns an Element object that represents the footer of the specified type for this section. The footer is returned as an editable element. There are 4 types of footers:
GetHeader(Section.HeaderFooterType)	Returns an Element object that represents the header of the specified type for this section. The header is returned as an editable element. There are 4 types of headers:

Nested Classes

Name	Description
------	-------------

BorderArea	Areas that a border can be applied to for a section.
BreakType	Types of breaks that can precede a section.
HeaderFooterType	Types of headers and footers.
LineNumbering	Line numbering settings for a section.
NumberFormat	Formats for page numbers in a section.
Orientation	Page orientations for a section.
PagePaperSize	Page paper sizes that can be applied to a section. All native sizes are supported, and additional sizes are provided