

# Creating Sections

A Section object represents a major section in a document, like a chapter in a book. Many documents will only contain one section. In more complex documents, content can be divided into multiple sections.

You can assign a different set of formatting settings to each section, including:

- Margins
- Paper size or orientation
- Paper source for a printer
- Page borders
- Vertical alignment
- Headers and footers
- Columns
- Page numbering
- Line numbering
- Footnotes and endnotes

To add a section to your document, use one of the following methods. The methods insert a section at the beginning or end an Element. An Element object represents an editable region in a document, such as a section or a paragraph. The document itself is an Element.

- `Section Element.CreateSectionAfter()`

Inserts an empty section at the end of the specified Element.

- `Section Element.CreateSectionBefore()`

Inserts an empty section at the beginning of the specified Element.

Before adding a section to your file, create a Document object.

```
WordApplication wvapp = new WordApplication();
Document doc = wvapp.Create();
```

Call `Document.CreateSectionAfter` or `Document.CreateSectionBefore` to add a section to the document

```
//--- Create a section at the end of the document.
Section firstSection = doc.CreateSectionAfter();
```

Next, set Section properties, for example:

```
//--- Add a footer to the section. A header or
//--- footer can be applied to all pages in a section,
//--- the first page in a section, odd pages only, or
//--- even pages only. The following returns a footer
//--- for all pages in the section.
Element footer = firstSection.GetFooter(Section.HeaderFooterType.All);
footer.InsertTextAfter("Copyright 2007 SoftArtisans Inc.", true);

//--- Set the width of a margin in twips (1/1440 in).
firstSection.LeftMargin = 1440;
firstSection.RightMargin = 1440;
```

After creating a Section and setting Section properties, add content. For example, add a paragraph:

```
Paragraph para = firstSection.InsertParagraphAfter
    (doc.Styles[NamedStyle.BuiltIn.Normal]);
para.InsertTextAfter("OfficeWriter is now fully integrated and compatible
    with SQL Server Reporting Services. Business users
    can now create and publish their Reporting Services
    reports without ever leaving Excel and Word. With
    OfficeWriter, every feature of Excel and Word is
    preserved in Reporting Services.",
    true);
```

You can access existing sections through the Document.Sections property.

```
//--- Open an existing Word file and get the first
//--- section.
WordApplication wwapp = new WordApplication();
Document doc = wwapp.Open(@"C:\sample.doc");
Section firstSection = doc.Sections[0];
```