Element

Description

This class is used to represent a region of the document that can be edited. This is the main class for manipulating the Word object model. Most elements of a Document inherit from the `Element` class.

C#

```
public class Element
```

vb.net

```
Public Class Element
```

Remarks

You cannot create a new `Element` object by using the new keyword. There are various properties

The following sample demonstrates getting an `Element` object by retrieving a section's header:

Examples

C#

```
WordApplication app = new WordApplication();
          Document doc = app.Open(@"C:\sample.doc");
          Section sect = doc.Sections[0];
          Element e = sect.GetHeader(Section.HeaderFooterType.All);
```

vb.net

```
Dim app As New WordApplication()
          Dim doc As Document = app.Open("C:\sample.doc")
          Dim sect As Section = doc.Sections(0)
          Dim e As Element = sect.GetHeader(Section.HeaderFooterType.All)
```

Properties

Name	Description
Children	Returns a collection of Element objects that represents all the direct children of the current element.

ElementType	Returns an Element.Type object that represents the type of this element.
Length	Returns an `int` that represents the number of characters in this Element.
Start	Returns an `int` that represents the absolute character index of the start of this Element.
Text	Returns a `String` that represents the text for this Element.

Methods

Name	Description
CreateBookmark(String)	Creates a bookmark on this Element with the specified name.
CreateSectionAfter()	Creates and returns a Section object that represents a new section at the end of this Element.
CreateSectionBefore()	Creates and returns a Section object that represents a new section at the beginning of this Element.
DeleteElement()	Deletes all the text and properties of this Element. The element then is no longer a valid element of the type. It is now just a position of where that element used to be.
GetElements(Element.Type)	Returns a collection of Element objects that represent the first level of elements in the document of the specified type. GetElements does not return nested elements.
GetPosition(Int32)	Returns a Position object that represents the location specified by the offset. The Position object can be used similar to a cursor.
ImportData(Object()())	Imports the data from a 2-dimensional array into a Table at the point of the current element in the document.
ImportData(Object()(), String(), DataImportProperties)	Imports a set of data from an array of column names and a 2-dimensional array of data to a Table created at the point of the current element in the document. Also accepts a DataImportProperties object defining any options for this data import.
ImportData(System.Data.DataTable)	Imports data from a `DataTable` into a Table at the point of this Element in the document.
ImportData(System.Data.DataTable, DataImportProperties)	Imports data from a `DataTable` into a Table at the point of this Element in the document. Also accepts a DataImportProperties object that contains the options for importing.
ImportData(System.Data.DataView)	Imports data from a `DataView` into a Table at the point of this Element in the document.
ImportData(System.Data.DataView, DataImportProperties)	Imports data from a `DataView` into a Table at the point of this Element in the document. Also accepts a DataImportProperties object that contains the options for importing.
ImportData(Object(,))	Imports the data from a rectangular array into a Table at the point of the current element in the document.
ImportData(Object(,), String(), DataImportProperties)	Imports the data from a rectangular array into a Table at the point of the current element in the document. Also accepts an array of column names and a DataImportProperties object with the import options.
ImportData(System.Data.IDataReader)	Imports the data from a DataReader into a Table at the point of the current element in the document.
ImportData(System.Data.IDataReader, DataImportProperties)	Imports the data from a DataReader into a Table at the point of the current element in the document. Also accepts a DataImportProperties object that contains the import options.

InsertAfter(Element)	Inserts an existing element after this element. A copy of the Element passed in is placed after this element
InsertBefore(Element)	Inserts an existing element before this element. A copy of the Element passed in is placed before this element
InsertBreakAfter(Element.BreakType)	Inserts a break of the specified type after the element.
InsertBreakBefore(Element.BreakType)	Inserts a break of the specified type before the element.
InsertHyperlinkAfter(String, String)	Creates and returns a Hyperlink at the beginning of this Element. This hyperlink will have the specified url and displayed text.
InsertHyperlinkBefore(String, String)	Creates and returns a Hyperlink at the beginning of this Element. This hyperlink will have the specified url and displayed text.
InsertImageAfter(System.IO.Stream)	Inserts and returns an InlineImage at the end of this Element. The image that will be inserted is specified by the stream parameter.
InsertImageAfter(String)	Inserts and returns an InlineImage at the end of this Element. The image that will be inserted is specified by the fileName parameter.
InsertImageBefore(String)	Inserts and returns an InlineImage at the beginning of this Element. The image that will be inserted is specified by the fileName parameter.
InsertImageBefore(System.IO.Stream)	Inserts and returns an InlineImage at the beginning of this Element. The image that will be inserted is specified by the stream parameter.
InsertListAfter(Boolean)	Creates and returns an empty List at the end of this Element. By passing a boolean, it can be a numbered list (true) or a bulleted list (false).
InsertListBefore(Boolean)	Creates and returns an empty List at the beginning of this Element. By passing a boolean, it can be a numbered list (true) or a bulleted list (false).
InsertMergeFieldAfter(String, String)	Creates and returns a MergeField at the end of this Element. This merge field will have the specified name and contents.
InsertMergeFieldBefore(String, String)	Creates and returns a MergeField at the beginning of this Element. This merge field will have the specified name and contents.
InsertParagraphAfter(NamedStyle)	Creates and returns a Paragraph object that represents a new empty paragraph at the end of this Element. The paragraph that is inserted will have the style specified. If this Element is in the middle of a paragraph, the containing paragraph will be split.
InsertParagraphAfter(NamedStyle, ParagraphFormatting)	Creates and returns a Paragraph object that represents a new empty paragraph at the end of this Element. The paragraph that is inserted will have the style specified, to which any additional specified paragraph formatting is applied as well. If this Element is in the middle of a paragraph, the containing paragraph will be split.
InsertParagraphBefore(NamedStyle)	Creates and returns a Paragraph object that represents a new empty paragraph at the beginning of this Element. The paragraph that is inserted will have the style specified. If this Element is in the middle of a paragraph, the containing paragraph will be split.
InsertParagraphBefore(NamedStyle, ParagraphFormatting)	Creates and returns a Paragraph object that represents a new empty paragraph at the beginning of this Element. The paragraph that is inserted will have the style specified, to which any additional specified paragraph formatting is applied as well. If this Element is in the middle of a paragraph, the containing paragraph will be split.
InsertTableAfter(Int32, Int32)	Creates and returns a Table at the end of this Element. The table will contain the specified number of rows and columns.
InsertTableAfter(Int32, Int32, TableFormatting)	Creates and returns a Table at the end of this Element. The table will contain the specified number of rows and columns. It will also be formatted with the properties specified.
InsertTableBefore(Int32, Int32)	Creates and returns a Table at the beginning of this Element. The table will contain the specified number of rows and columns.

InsertTableBefore(Int32, Int32, TableFormatting)	Creates and returns a Table at the beginning of this Element. The table will contain the specified number of rows and columns. It will also be formatted with the properties specified.
InsertTextAfter(String, Boolean)	Creates and returns a CharacterRun at the end of this Element. The text to be inserted is specified as well as whether this should be a new character run or part of the previous one.
InsertTextAfter(String, Font)	Creates and returns a CharacterRun at the end of this Element. The text to be inserted is specified as well as the font that should override the font found in the style of the character run's enclosing paragraph.
InsertTextBefore(String, Boolean)	Creates and returns a CharacterRun at the beginning of this Element. The text to be inserted is specified as well as whether this should be a new character run or part of the existing one.
InsertTextBefore(String, Font)	Creates and returns a CharacterRun at the beginning of this Element. The text to be inserted is specified as well as the font that should override the font found in the style of the character run's enclosing paragraph.
Search(String)	Searches for a specified string and returns a collection of type SearchMatch[].
SearchAndReplace(String, String)	Searches for a string specified by the `search` parameter, and replaces it with the string specified by the `replaceWith` parameter. The search string can be a literal, a regular expression, or a combination of both. Since the string can contain regular expressions, use a backslash to escape special characters. Special characters are `[^$.|?*+()`.
SubElement(Int32, Int32)	Creates and returns a Element object that represents a sub Element within this Element based on the specified start and end offsets.

Extension Methods

Overload	Description
ImportData(Microsoft.SharePoint.SPList)	Imports the data from a SharePoint List into a Table at the point of the current element in the document.
ImportData(Microsoft.SharePoint.SPView, Microsoft.SharePoint.SPList, DataImportProperties)	Imports the data from a SharePoint View into a Table at the point of the current element in the document. Also accepts a SharePoint List to which the View belongs and a DataImportProperties object with the import options.

Nested Classes

Name	Description
BreakType	An Element.BreakType value specifies the type of break to insert Element.InsertBreakBefore or Element.InsertBreakAfter a region in the document.
Type	An Element represents a region of the document that can be edited. An Element.Type value specifies the type of region an `Element` object represents.