

# Creating Lists

## Lists

To add a list to a section of your document, use one of the following methods. All of the methods insert a list at a specified point in an Element. An Element object represents an editable region in a document, such as a section or a table cell. The document itself is an Element (the Document class extends Element). Later, we will add a list to a Document object.

- `List Element.InsertListAfter(boolean numbered)`

Inserts an empty list at the end of the specified Element. Pass true to the method to create a numbered list, and false to create a bulleted list.

- `List Element.InsertListBefore(boolean numbered)`

Inserts an empty list at the beginning of the specified Element. Pass true to the method to create a numbered list, and false to create a bulleted list.

- `List Position.InsertList(boolean numbered)`

Inserts an empty list at a specified position within an Element. A Position object represents a cursor. Pass true to the method to create a numbered list, and false to create a bulleted list.

Before adding a list to your file, you must create a content region in which to insert the list. Each editable region in a Word file is represented by an Element object, or an object that extends Element, such as a Document object.

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

To insert an empty list, use one of the methods listed at the beginning of this section, for example:

```
//--- The boolean parameter specifies whether the
//--- list is numbered (true) or bulleted (false).
List numberedList = doc.InsertListAfter(true);
```

Next, create list entries. To add a new entry, call either `List.AddEntry` or `List.InsertEntry`.

```
//--- AddEntry's parameter specifies the 0-based
//--- indent level at which to insert the new entry. A list
//--- may contain up to 9 levels, so the deepest level
//--- is 8. The following inserts the first list entry, at
//--- the top list level.
ListEntry firstListEntry = numberedList.AddEntry(0);

//--- InsertEntry's first parameter specifies the 0-based
//--- position in the list at which to insert the new entry
//--- (for the first entry this parameter should be 0,
//--- for the second 1, and so on). The second parameter
//--- specifies the 0-based level of the new entry. A list
//--- may contain up to 9 levels, so the deepest level
//--- is 8. The following inserts the second list entry, at
//--- the top list level.
ListEntry secondListEntry = numberedList.InsertEntry(1, 0);
```

To add text to a list entry call one of the `Element` class's `InsertTextAfter` or `InsertTextBefore` methods. You can [apply a font](#) to the text by passing a `Font` object to `InsertTextAfter` or `InsertTextBefore`. The following lines create a `Font` object that we will use later when adding text to the paragraph:

```
Font listFont = doc.CreateFont();
listFont.FontName = "Times New Roman";
listFont.FontSize = 10;
```

The method `InsertTextAfter` takes a string and inserts it at the end of the list entry. The method's second parameter specifies a `Font` object (created above) to apply to the text.

```
firstListEntry.InsertTextAfter("OfficeWriter for Word",
    listFont);
secondListEntry.InsertTextAfter("OfficeWriter for Excel",
    listFont);
```

You can access existing lists through the `Element.Elements` property.

```
//--- Open an existing Word file and get the first
//--- table.
WordApplication wwapp = new WordApplication();
Document doc = wwapp.Open(@"C:\sample.doc");
List firstList = doc.GetElements(Element.Type.List)[0];
```

## See Also

[List](#)

[ListEntry](#)

[ListLevel](#)