

Creating Tables

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

- `Table Element.InsertTableAfter(int rows, int columns)`

Inserts an empty table, with a specified number of rows and columns, at the end of the element.

- `Table Element.InsertTableAfter(int rows, int columns, TableFormatting props)`

Inserts an empty table, with a specified number of rows and columns, at the end of the element. This method applies a set of table formatting properties (contained in a TableFormatting object) to the new table.

- `Table Element.InsertTableBefore(int rows, int columns)`

Inserts an empty table, with a specified number of rows and columns, at the beginning of the element.

- `Table Element.InsertTableBefore(int rows, int columns, TableFormatting props)`

Inserts an empty table, with a specified number of rows and columns, at the beginning of the element. This method applies a set of table formatting properties (contained in a TableFormatting object) to the new table.

- `Table Position.InsertTable(int rows, int columns)`

Inserts an empty table at a specified position within an Element. A Position object represents a cursor.

Before adding a table to your file, you must create a content region in which to insert the table. 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 wwapp = new WordApplication();
Document doc = wwapp.Create();
```

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

```
//--- Create a table with 5 rows and 4 columns
//--- and insert it at the end of the document.
Table tbl = doc.InsertTableAfter(5, 4);
```

To enter text in the table, first return one or more TableCell objects:

```
TableCell firstCell = tbl[0, 0];
TableCell secondCell = tbl[0, 1];
```

To add text to a table cell, call one of the Element class's InsertTextAfter or InsertTextBefore methods.

```
//--- When adding text to a table, you can
//--- specify a font for the text. First, create a Font
//--- object:
Font cellFont = doc.CreateFont();
cellFont.FontName = "Times New Roman";
cellFont.FontSize = 10;

//--- The following method takes a string and inserts it
//--- at the end of the table cell. The method's second
//--- parameter specifies a Font object to apply to the
//--- text.
firstCell.InsertTextAfter("OfficeWriter for Word",
    cellFont);
secondCell.InsertTextAfter("OfficeWriter for Excel",
    cellFont);
```

You can access existing tables 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");
Table firstTable = (Table)doc.GetElements(Element.Type.Table)[0];
```

Table Formatting

To apply a font to text in a table cell, create a `Font` object and pass it to `TableCell.InsertTextAfter` or `TableCell.InsertTextBefore`, as demonstrated above.

Other paragraph formatting can be applied through the `TableObject` or the `TableFormatting` object. To create a `TableFormatting` object, call `Document.CreateTableFormatting`:

```
WordApplication wwapp = new WordApplication();
Document doc = wwapp.Create();
TableFormatting tblFormatting = doc.CreateTableFormatting();
```

Next, set `TableFormatting` properties.

```
tblFormatting.DefaultShading.BackgroundColor = Color.Gray;
tblFormatting.set_DefaultPadding(TableCell.Location.Top, 720);
```

To assign the `TableFormatting` object that you created to a table, pass it to `Element.InsertTableAfter` or `Element.InsertTableBefore` when you create a new table.

```
Table tbl = doc.InsertTableAfter(3, 5, tblFormatting);
```