

Table

Description

Represents a table in the Word document.

C#

```
[DefaultMember("Item")]  
public sealed class Table : Element
```

vb.net

```
<DefaultMember("Item")> _  
Public NotInheritable Class Table  
    Inherits Element
```

Remarks

To create a new table, use [Element.InsertTableBefore\(\)](#) or [InsertTableAfter\(\)](#). To get an existing table, use [Element.GetElements](#) or [Element.Children](#) and check which elements are of type [Element.Type.Table](#).

The following sample demonstrates inserting a new table at the end of a document as well as finding the first table in a document.

Examples

C#

```
//--- Insert a table at the end of a new document with 3 rows and 5 columns  
WordApplication app = new WordApplication();  
Document doc = app.Create();  
Table table = doc.InsertTableAfter(3, 5);  
  
//--- Get the first table of an existing document  
WordApplication app = new WordApplication();  
Document doc = app.Open(@"C:\sample.doc");  
Table table = doc.GetElements(Element.Type.Table)[0];
```

vb.net

```
'--- Insert a table at the end of a new document with 3 rows and 5 columns
Dim app As New WordApplication()
Dim doc As Document = app.Create()
Dim table As Table = doc.InsertTableAfter(3, 5)

'--- Get the first table of an existing document
Dim app As New WordApplication()
Dim doc As Document = app.Open("C:\sample.doc")
Dim table As Table = doc.GetElements(Element.Type.Table)(0)
```

Properties

Name	Description
AbsolutePositioning	Returns an AbsolutePositioning object on which you can control the absolute positioning for the table.
Formatting	Sets or returns a TableFormatting object representing the formatting of this table (borders, padding, shading, etc).
NumRows	Returns an <code>int</code> representing the number of rows in the first row of this table.
PreferredWidth	The page <code>Table.PreferredWidth</code> could not be found.
Style	Sets or returns a NamedStyle object that represents the Style of the current table.

Indexers

Name	Description
Item(Int32, Int32)	Returns a TableCell object that represents the table cell that exists at the given column and row.

Methods

Name	Description
AddColumns(Int32)	Adds a specified number of columns to the horizontal end of the table.
AddRows(Int32)	Adds a specified number of rows to the vertical end of the table.
CreateBookmarkOnRow(Int32, String)	Creates a bookmark on the specified row using the specified name.
DeleteColumns(Int32, Int32)	{excerpt-include:Table.DeleteColumns(Int32,
DeleteRows(Int32, Int32)	Deletes one or more rows from the table.
GetNumColumns(Int32)	Returns the number of columns in the specified row of the table.
GetPreferredColumnWidth(Int32)	Returns an <code>int</code> representing the width of a particular column. The GetPreferredColumnWidth method returns the width of the column in twips. One twip = (1/20 pt) or (1/1440 in).

<code>GetPreferredColumnWidthUnits(Int32)</code>	Returns a TableCell.WidthUnits object that represents the units used for the width of a particular column.
<code>GetPreferredWidth(out Units)</code>	Returns a double that represents the preferred width of this cell, with the associated units return in units .
<code>GetRowHeight(Int32)</code>	Returns an <code>int</code> representing the height of a particular row in the table. The default row height unit is twips. One twip = (1/20 pt) or (1/1440 in).
<code>GetRowHeightExact(Int32)</code>	Returns a <code>boolean</code> that represents if the row height is "exact" (true) or "at least" (false).
<code>ImportDataRow(Object[])</code>	Imports data from an array of objects to the current table.
<code>InsertColumns(Int32, Int32)</code>	Inserts one or more columns at the specified position in the table.
<code>InsertRows(Int32, Int32)</code>	Inserts one or more rows at the specified position in the table.
<code>MergeCells(Int32, Int32, Int32, Int32)</code>	Merges the specified set of cells into a single cell.
<code>SetPreferredColumnWidth(Int32, Int32)</code>	Sets an <code>int</code> representing the width of a particular column. In order to set the width of a column, you must first set the PreferredColumnWidthUnit of that column.
<code>SetPreferredColumnWidthUnits(Int32, TableCell.WidthUnits)</code>	Sets a TableCell.WidthUnits object that represents the units used for the width of a particular column.
<code>SetPreferredWidth(Double, Units)</code>	Sets the preferred width of the table and the units for said width.
<code>SetRowHeight(Int32, Int32)</code>	Sets an <code>int</code> representing the height of a particular row in the table. A row height is set in twips. One twip = (1/20 pt) or (1/1440 in).
<code>SetRowHeightExact(Int32, Boolean)</code>	Sets a <code>boolean</code> that represents if the row height is "exact" (true) or "at least" (false).