

# How to: Programmatically Create Word Tables

The [Tables](#) collection is a member of the [Microsoft.Office.Interop.Word.Document](#), [Microsoft.Office.Tools.Word.Document](#), [Selection](#), and [Range](#) classes, which means that you can create a table in any of those contexts. You use the [M:Microsoft.Office.Interop.Word.Tables.Add\(Microsoft.Office.Interop.Word.Range,System.Int32,System.Int32,System.Object@,System.Object@\)](#) method of the [Tables](#) collection to add a table at the specified range.

**Applies to:** The information in this topic applies to document-level projects and VSTO add-in projects for Word. For more information, see [Features Available by Office Application and Project Type](#).

## Creating Tables in Document-Level Customizations

### To add a simple table to a document

- Use the [M:Microsoft.Office.Interop.Word.Tables.Add\(Microsoft.Office.Interop.Word.Range,System.Int32,System.Int32,System.Object@,System.Object@\)](#) method to add a table consisting of three rows and four columns at the beginning of the document.

To use the following code example, run it from the `ThisDocument` class in your project.

**C#**

```
object start = 0;
object end = 0;
Word.Range tableLocation = this.Range(ref start, ref end);
this.Tables.Add(tableLocation, 3, 4);
```

When you create a table, it is automatically added to the [Tables](#) collection of the [Microsoft.Office.Tools.Word.Document](#) host item. You can then refer to the table by its item number by using the [P:Microsoft.Office.Interop.Word.Tables.Item\(System.Int32\)](#) property, as shown in the following code.

### To refer to a table by item number

- Use the [P:Microsoft.Office.Interop.Word.Tables.Item\(System.Int32\)](#) property and supply the item number of the table that you want to refer to.

To use the following code example, run it from the `ThisDocument` class in your project.

**C#**

```
Word.Table newTable = this.Tables[1];
```

Each [Table](#) object also has a [P:Microsoft.Office.Interop.Word.Table.Range](#) property that enables you to set formatting attributes.

### To apply a style to a table

1. Use the [P:Microsoft.Office.Interop.Word.Table.Style](#) property to apply one of the Word built-in styles to a table.

To use the following code example, run it from the `ThisDocument` class in your project.

**C#**

```
this.Tables[1].Range.Font.Size = 8;  
this.Tables[1].set_Style("Table Grid 8");
```

## Creating Tables in VSTO Add-ins

### To add a simple table to a document

- Use the [M:Microsoft.Office.Interop.Word.Tables.Add\(Microsoft.Office.Interop.Word.Range,System.Int32,System.Int32,System.Object@,System.Object@\)](#) method to add a table consisting of three rows and four columns at the beginning of the document.

The following code example adds a table to the active document. To use this example, run it from the `ThisAddIn` class in your project.

**C#**

```
Word.Range tableLocation =  
    this.Application.ActiveDocument.Range(0, 0);  
this.Application.ActiveDocument.Tables.Add(  
    tableLocation, 3, 4);
```

When you create a table, it is automatically added to the [Tables](#) collection of the [Microsoft.Office.Interop.Word.Document](#). You can then refer to the table by its item number by using the [P:Microsoft.Office.Interop.Word.Tables.Item\(System.Int32\)](#) property, as shown in the following code.

### To refer to a table by item number

1. Use the [P:Microsoft.Office.Interop.Word.Tables.Item\(System.Int32\)](#) property and supply the item number of the table that you want to refer to.

The following code example uses the active document. To use this example, run it from the `ThisAddIn` class in your project.

**C#**

```
Word.Table newTable = this.Application.ActiveDocument.Tables[1];
```

Each [Table](#) object also has a [P:Microsoft.Office.Interop.Word.Table.Range](#) property that enables you to set formatting attributes.

## To apply a style to a table

1. Use the [P:Microsoft.Office.Interop.Word.Table.Style](#) property to apply one of the Word built-in styles to a table.

The following code example uses the active document. To use this example, run it from the `ThisAddIn` class in your project.

**C#**

```
this.Application.ActiveDocument.Tables[1].Range.Font.Size = 8;  
this.Application.ActiveDocument.Tables[1].set_Style("Table Grid 8");
```

## See Also

[How to: Programmatically Add Text and Formatting to Cells in Word Tables](#)

[How to: Programmatically Add Rows and Columns to Word Tables](#)

[How to: Programmatically Populate Word Tables with Document Properties](#)

[Optional Parameters in Office Solutions](#)

© 2017 Microsoft