How to: Programmatically Insert Text into Word Documents

Other Versions ▾

There are three primary ways to insert text into Microsoft Office Word documents:

- Insert text in a range.
- Replace text in a range with new text.
- Use the M:Microsoft.Office.Interop.Word.Selection.TypeText(System.String) method of a Selection object to insert text at the cursor or selection.

> **Note**
>
> You can also insert text into content controls and bookmarks. For more information, see Content Controls and Bookmark Control.

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.

Inserting Text in a Range

Use the P:Microsoft.Office.Interop.Word.Range.Text property of a Range object to insert text in a document.

To insert text in a range

1. Specify a range at the beginning of a document and insert the text **New Text**.

   The following code example can be used in a document-level customization.

   C# **VB**

```
object start = 0;
object end = 0;

Word.Range rng = this.Range(ref start, ref end);
rng.Text = "New Text";
```

The following code example can be used in a VSTO Add-in. This code uses the active document.

C# VB

```
Word.Range rng = this.Application.ActiveDocument.Range(0, 0);
rng.Text = "New Text";
```

2. Select the Range object, which has expanded from one character to the length of the inserted text.

C# VB

```
rng.Select();
```

Replacing Text in a Range

If the specified range contains text, all text in the range is replaced with the inserted text.

To replace text in a range

1. Create a Range object that consists of the first 12 characters in the document.

   The following code example can be used in a document-level customization.

C# VB

```
object start = 0;
object end = 12;

Word.Range rng = this.Range(ref start, ref end);
```

The following code example can be used in a VSTO Add-in. This code uses the active document.

C# VB

```
Word.Range rng = this.Application.ActiveDocument.Range(0, 12);
```

2. Replace those characters with the string **New Text**.

C# VB

```
rng.Text = "New Text";
```

3. Select the range.

C# VB

```
            rng.Select();
```

Inserting Text Using TypeText

The M:Microsoft.Office.Interop.Word.Selection.TypeText(System.String) method inserts text at the selection. M:Microsoft.Office.Interop.Word.Selection.TypeText(System.String) behaves differently depending on the options set on the user's computer. The code in the following procedure declares a Selection object variable, and turns off the **Overtype** option if it is turned on. If the **Overtype** option is activated, then any text next to the cursor is overwritten.

To insert text using the TypeText method

1. Declare a Selection object variable.

C# VB

```
            Word.Selection currentSelection = Application.Selection;
```

2. Turn off the **Overtype** option if it is turned on.

C# VB

```
            if (Application.Options.Overtype)
            {
                Application.Options.Overtype = false;
            }
```

3. Test to see whether the current selection is an insertion point.

   If it is, the code inserts a sentence using M:Microsoft.Office.Interop.Word.Selection.TypeText(System.String), and then a paragraph mark using the M:Microsoft.Office.Interop.Word.Selection.TypeParagraph method.

C# VB

```
            // Test to see if selection is an insertion point.
            if (currentSelection.Type == Word.WdSelectionType.wdSelectionIP)
            {
                currentSelection.TypeText("Inserting at insertion point. ");
                currentSelection.TypeParagraph();
            }
```

4. The code in the **ElseIf** block tests to see whether the selection is a normal selection. If it is, then another **If** block tests to see whether the **ReplaceSelection** option is turned on. If it is, the code uses

the M:Microsoft.Office.Interop.Word.Selection.Collapse(System.Object@) method of the selection to collapse the selection to an insertion point at the start of the selected block of text. Insert the text and a paragraph mark.

C# VB

```
            else
                if (currentSelection.Type == Word.WdSelectionType.wdSelectionNormal)
                {
                    // Move to start of selection.
                    if (Application.Options.ReplaceSelection)
                    {
                        object direction = Word.WdCollapseDirection.wdCollapseStart;
                        currentSelection.Collapse(ref direction);
                    }
                    currentSelection.TypeText("Inserting before a text block. ");
                    currentSelection.TypeParagraph();
                }
```

5. If the selection is not an insertion point or a block of selected text, then the code in the **Else** block does nothing.

C# VB

```
                else
                {
                    // Do nothing.
                }
```

You can also use the M:Microsoft.Office.Interop.Word.Selection.TypeBackspace method of the Selection object, which mimics the functionality of the BACKSPACE key on your keyboard. However, when it comes to inserting and manipulating text, the Range object offers you more control.

The following example shows the complete code. To use this example, run the code from the `ThisDocument` or `ThisAddIn` class in your project.

C# VB

```
        private void SelectionInsertText()
        {
            Word.Selection currentSelection = Application.Selection;

            // Store the user's current Overtype selection
            bool userOvertype = Application.Options.Overtype;

            // Make sure Overtype is turned off.
            if (Application.Options.Overtype)
            {
                Application.Options.Overtype = false;
            }

            // Test to see if selection is an insertion point.
            if (currentSelection.Type == Word.WdSelectionType.wdSelectionIP)
            {
                currentSelection.TypeText("Inserting at insertion point. ");
                currentSelection.TypeParagraph();
            }
            else
                if (currentSelection.Type == Word.WdSelectionType.wdSelectionNormal)
                {
                    // Move to start of selection.
                    if (Application.Options.ReplaceSelection)
                    {
                        object direction = Word.WdCollapseDirection.wdCollapseStart;
                        currentSelection.Collapse(ref direction);
                    }
                    currentSelection.TypeText("Inserting before a text block. ");
                    currentSelection.TypeParagraph();
                }
                else
                {
                    // Do nothing.
                }

            // Restore the user's Overtype selection
            Application.Options.Overtype = userOvertype;
        }
```