How to: Programmatically Loop Through Found Items in Documents

The Find class has a P:Microsoft.Office.Interop.Word.Find.Found property, which returns **true** whenever a searched-for item is found. You can loop through all instances found in a Range using the M:Microsoft.Office.Interop.Word.Find.Execute(System.Object@,System.Object@,System.Object@,System.Object@,System.Object @,System.Object@,System.Object@,System.Object@,System.Object@,System.Object@,System.Object@,System.Object@,System. Object@,System.Object@,System.Object@) method.

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.

To loop through found items

1. Declare a Range object.

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

   ```csharp
   C#

               Word.Range rng = this.Content;
   ```

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

   ```csharp
   C#

               Word.Document document = this.Application.ActiveDocument;
               Word.Range rng = document.Content;
   ```

2. Use the P:Microsoft.Office.Interop.Word.Find.Found property in a loop to search for all occurrences of the string in the document, and increment an integer variable by 1 each time the string is found.

   ```csharp
   C#

               rng.Find.ClearFormatting();
               rng.Find.Forward = true;
               rng.Find.Text = "find me";

               rng.Find.Execute(
                   ref missing, ref missing, ref missing, ref missing, ref missing,
                   ref missing, ref missing, ref missing, ref missing, ref missing,
                   ref missing, ref missing, ref missing, ref missing, ref missing);
   ```

```csharp
            while (rng.Find.Found)
            {
                intFound++;
                rng.Find.Execute(
                    ref missing, ref missing, ref missing, ref missing, ref missing,
                    ref missing, ref missing, ref missing, ref missing, ref missing,
                    ref missing, ref missing, ref missing, ref missing, ref missing);
            }
```

3. Display the number of times the string was found in a message box.

C#

```csharp
            MessageBox.Show("Strings found: " + intFound.ToString());
```

The following examples show the complete method.

Document-Level Customization Example

To loop through items in a document-level customization

1. The following example shows the complete code for a document-level customization. To use this code, run it from the
   ThisDocument class in your project.

C#

```csharp
        private void FindLoop()
        {
            int intFound = 0;
            Word.Range rng = this.Content;

            rng.Find.ClearFormatting();
            rng.Find.Forward = true;
            rng.Find.Text = "find me";

            rng.Find.Execute(
                ref missing, ref missing, ref missing, ref missing, ref missing,
                ref missing, ref missing, ref missing, ref missing, ref missing,
                ref missing, ref missing, ref missing, ref missing, ref missing);

            while (rng.Find.Found)
            {
                intFound++;
                rng.Find.Execute(
                    ref missing, ref missing, ref missing, ref missing, ref missing,
                    ref missing, ref missing, ref missing, ref missing, ref missing,
                    ref missing, ref missing, ref missing, ref missing, ref missing);
            }
```

```
            MessageBox.Show("Strings found: " + intFound.ToString());
        }
```

VSTO Add-in Example

To loop through items in an VSTO Add-in

1. The following example shows the complete code for an VSTO Add-in. To use this code, run it from the `ThisAddIn` class in your project.

C#

```csharp
private void FindLoop()
{
    int intFound = 0;
    Word.Document document = this.Application.ActiveDocument;
    Word.Range rng = document.Content;

    rng.Find.ClearFormatting();
    rng.Find.Forward = true;
    rng.Find.Text = "find me";

    rng.Find.Execute(
        ref missing, ref missing, ref missing, ref missing, ref missing,
        ref missing, ref missing, ref missing, ref missing, ref missing,
        ref missing, ref missing, ref missing, ref missing, ref missing);

    while (rng.Find.Found)
    {
        intFound++;
        rng.Find.Execute(
            ref missing, ref missing, ref missing, ref missing, ref missing,
            ref missing, ref missing, ref missing, ref missing, ref missing,
            ref missing, ref missing, ref missing, ref missing, ref missing);
    }

    MessageBox.Show("Strings found: " + intFound.ToString());
}
```

See Also

How to: Programmatically Search for and Replace Text in Documents
How to: Programmatically Set Search Options in Word
How to: Programmatically Define and Select Ranges in Documents

How to: Programmatically Restore Selections After Searches

Optional Parameters in Office Solutions

© 2017 Microsoft

How to: Programmatically Restore Selections After Searches

Optional Parameters in Office Solutions