

# HTMLToWord.InsertDelegate

## Description

A user-assigned delegate method that HTMLToWord calls before it processes an XHTML tag. The user may implement code to perform any tasks they desire before passing control back to HTMLToWord. The return value from the delegate method specifies how HTMLToWord should behave. If this property is not set, HTMLToWord will attempt to insert the contents of the XHTML tag.

**C#**

```
public InsertElementDelegate InsertDelegate{ get; set; }
```

**vb.net**

```
Public Property InsertDelegate() As InsertElementDelegate
```

## Examples

## C#

```
//--- We have defined two custom HTML tags for our documents: para and private.
//--- If we encounter a <para> tag, we insert a paragraph into the Word document
and tell HTMLToWord
//--- to continue processing. If we encounter a <private> tag, we write a short
message into
//--- the Word document and tell HTMLToWord to skip this tag altogether.
HTMLToWord.HTMLTagAction MyInserter(Document doc, Element insertAfterElement, XmlNode
node)
{
    //--- Get the name of the HTML tag and convert it to lower case.
    string nodeName = node.Name.ToLower();

    if (nodeName.Equals("para"))
    {
        //--- There is a special <para> tag in our markup, but we should just
        treat it
        //--- like a standard HTML <p> tag. Insert a paragraph into the
        document and
        //--- let HTML continue processing.
        insertAfterElement.InsertParagraphAfter(null);
    }
    else if (nodeName.Equals("private"))
    {
        //--- The "private" tag is used to indicate information that should not be
        sent
        //--- to outside clients; therefore, we tell HTMLToWord to skip this tag
        and its
        //--- contents.
        return HTMLToWord.HTMLTagAction.Skip;
    }

    //--- We have handled all of the special cases. Let HTMLToWord know that
    //--- it should process this tag.
    return HTMLToWord.HTMLTagAction.Process;
}

//--- Create an HTMLToWord instance.
HTMLToWord h2w = new HTMLToWord();

//--- Tell HTMLToWord that it should call our "MyInserter" delegate method
//--- whenever it encounters a tag in the XHTML string.
h2w.InsertDelegate = new HTMLToWord.InsertElementDelegate(MyInserter);
```

## vb.net

```
'--- We have defined two custom XHTML tags for our documents: para and private.
'--- If we encounter a <para> tag, we insert a paragraph into the Word document
and tell HTMLToWord
'--- to continue processing. If we encounter a <private> tag, we write a short
message into
'--- the Word document and tell HTMLToWord to skip this tag altogether.
Function MyInserter(ByVal doc As Document, ByVal insertAfterElement As Element, ByVal
xmlNode As XmlNode)
    As HTMLToWord.HTMLTagAction
{
    '--- Get the name of the HTML tag and convert it to lower case.
    Dim nodeName As String = node.Name.ToLower()

    If nodeName.Equals("para") Then
        '--- There is a special <para> tag in our markup, but we should just
treat it
        '--- like a standard HTML <p> tag. Insert a paragraph into the document
and
        '--- let HTML continue processing.
        insertAfterElement.InsertParagraphAfter(Nothing)
    Else If nodeName.Equals("private")
        '--- The "private" tag is used to indicate information that should not be
sent
        '--- to outside clients; therefore, we tell HTMLToWord to skip this tag and
its
        '--- contents.
        Return HTMLToWord.HTMLTagAction.Skip;
    End If

    '--- We have handled all of the special cases. Let HTMLToWord know that
    '--- it should process this tag.
    Return HTMLToWord.HTMLTagAction.Process
End Function

'--- Create an HTMLToWord instance.
Dim h2w As New HTMLToWord()

'--- Tell HTMLToWord that it should call our "MyInserter" delegate method
'--- whenever it encounters a tag in the XHTML string.
h2w.InsertDelegate = New HTMLToWord.InsertElementDelegate(AddressOf MyInserter)
```