

# La documentation du code en C#

(extrait du site de JM Doudoux <http://www.jmdoudoux.fr/microsoft/doctechcsharp/doctechcsharp.htm>)

## La documentation du code en C#

La documentation du code est particulièrement utile lors des évolutions et de la maintenance du code. C# propose un mécanisme simple pour insérer dans le code des commentaires de documentation qui seront traités pour réaliser une documentation technique.

Ces commentaires de documentation sont préfixés par ///

Ces commentaires sont saisis avant la déclaration des entités de type namespace, classe, interface, champ, propriété, méthode, ou événement.

Ces commentaires peuvent utiliser des tags XML pour caractériser certaines informations. Ces tags seront repris dans un document XML peut être généré à la demande par le compilateur. Il est donc important que les commentaires de documentation saisis respectent le standard XML.

Ce fichier XML peut ensuite être traité pour par exemple produire des pages HTML grâce à une feuille de style XSLT.

Microsoft a défini une liste de tags particuliers recommandés mais il est tout à fait possible d'utiliser ces propres tags.

## Les tags de documentation recommandés

Les tags XML suivants sont utilisables pour la documentation du code :

<c>	<code>	<example>
<exception>	<include>	<list>
<para>	<param>	<paramref>
<permission>	<remarks>	<return>
<see>	<seealso>	<summary>
<value>		

Ces tags sont utilisables en fonction de l'entité documentée

Entité	Tags utilisables
class	<summary>, <remarks>, <seealso>
struct	<summary>, <remarks>, <seealso>
interface	<summary>, <remarks>, <seealso>
delegate	<summary>, <remarks>, <seealso>, <param>, <returns>
enum	<summary>, <remarks>, <seealso>
constructor	<summary>, <remarks>, <seealso>, <param>, <permission>, <exception>
property	<summary>, <remarks>, <seealso>, <value>, <permission>, <exception>
method	<summary>, <remarks>, <seealso>, <param>, <returns>, <permission>, <exception>
event	<summary>, <remarks>, <seealso>

## Commentaires sur les principaux tags utilisables

### Le tag <c>

Ce tag permet d'afficher le texte qu'il contient sous la forme de code

Syntaxe :

<c>text</c>

Exemple :

/// resume de <c>maMethode</c>

## Le tag <code>

Ce tag permet d'afficher le texte qu'il contient sous la forme de code. La différence avec le tag <c> est que le tag <code> est utilisé en multi-ligne.

Syntaxe :

```
<code>
```

```
ligne1
```

```
ligne2
```

```
</code>
```

Exemple

```
/// <code>
```

```
/// Class1 m = new Class1();
```

```
/// m.maMethode("Hello");
```

```
/// </code>
```

## Le tag <example>

Ce tag permet de fournir un exemple d'utilisation de l'entité.

Syntaxe :

```
<example>
```

```
texte
```

```
</example>
```

Exemple :

```
/// <example>
```

```
/// Mise en oeuvre :
```

```
/// <code>
```

```
/// Class1 m = new Class1();
```

```
/// m.maMethode("Hello");
```

```
/// </code>
```

```
/// </example>
```

### **Le tag <exception>**

Ce tag permet de fournir des informations sur la levée d'une exception par une méthode

### **Le tag <list>**

Ce tag permet de créer une liste d'élément avec puce, ou numéroté ou sous forme de tableau.

Syntaxe :

```
<list type="bullet" | "number" | "table">
  <listheader>
    <term>term</term>
    <description>description</description>
  </listheader>
  <item>
    <term>term</term>
    <description>description</description>
  </item>
</list>
```

### **Le tag <para>**

Ce tag permet de créer un paragraphe à l'intérieur d'un tag.

Syntaxe

```
<para>text</para>
```

### Le tag <param>

Ce tag permet de fournir des informations sur un paramètre d'une méthode.

Syntaxe :

```
<param name='name'>description</param>
```

L'attribut name permet de préciser le nom du paramètre.

Exemple :

```
/// <param name="nom">nom de l'utilisateur</param>
```

### Le tag <paramref>

Ce tag permet de faire référence à un paramètre dans le texte

Syntaxe :

```
<paramref name="name"/>
```

### Le tag <remarks>

Ce tag permet de fournir des informations complémentaires sur une entité sous la forme d'une remarque.

Syntaxe :

```
<remarks>description</remarks>
```

Exemple :

```
/// <remarks>remarque complémentaire.</remarks>
```

### Le tag <return>

Ce tag permet de fournir de informations sur la valeur de retour d'une méthode

Syntaxe :

```
<returns>description</returns>
```

Exemple :

```
/// <returns>une chaine contenant une salutation</returns>
```

### Le tag <see>

Ce tag permet de faire un lien vers un élément accessible dans le code.

Syntaxe :

```
<see cref="member"/>
```

L'attribut cref permet de préciser le membre.

Exemple :

```
/// <see cref="maMethode" />
```

Le compilateur vérifie l'accessibilité du membre précisé dans l'attribut cref : si celle-ci échoue, il émet un avertissement

Exemple :

Le commentaire XML sur 'ClassLibrary.Class1.maMethode(string)' possède l'attribut

cref 'maMethode2df' et celui-ci est introuvable

### Le tag <seealso>

Ce tag permet de faire un lien vers un élément qui sera inclus dans la section see also.

Syntaxe :

```
<seealso cref="member"/>
```

L'attribut cref permet de préciser le membre.

Exemple :

```
/// <seealso cref="ClassLibrary.MaClasse2"/>
```

### Le tag <summary>

Ce tag permet de fournir un résumé d'une entité.

Syntaxe

```
<summary>description</summary>
```

Exemple :

```
/// <summary>
```

```
/// resume de maMethode
```

```
/// </summary>
```

Le contenu de ce tag est utilisé par la fonction Intellisens de Visual Studio .Net pour afficher des informations sur l'entité.

### Le tag <value>

Ce tag permet de fournir des informations sur une propriété

Syntaxe :

<value>description</value>

Exemple :

```
/// <value> valeur entiere utilise dans les calculs
```

```
/// </value>
```

## Mise en œuvre avec Visual Studio

### Saisie des commentaires de documentation

```
17 public string maMethode(string nom)
18 {
19     return "Bonjour " + nom;
20 }
```

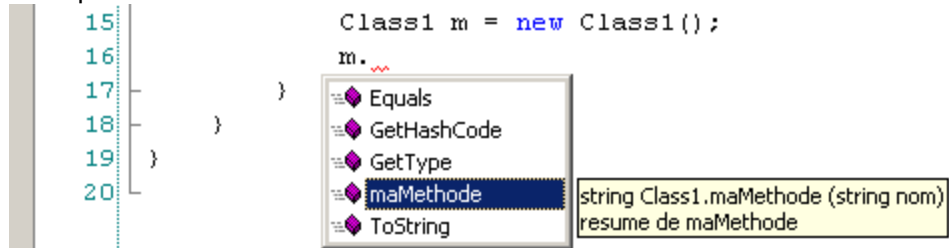
Il suffit de saisir /// sur la ligne avant la déclaration de l'entité pour que Visual Studio .Net génère automatiquement un squelette de commentaire de documentation en tenant compte de la déclaration de l'entité.

```
17 /// <summary>
18 /// |
19 /// </summary>
20 /// <param name="nom"></param>
21 /// <returns></returns>
22 public string maMethode(string nom)
23 {
24     return "Bonjour " + nom;
25 }
```

## Utilisation automatique de la documentation par Visual Studio

Ce fichier va être utilisé par Visual Studio .Net pour fournir des informations dans l'éditeur de code.

Exemple : lors de la sélection d'une méthode



Exemple : lors de sélection de la saisie d'un paramètre

