

Date: Vendredi 28 septembre 2012 &agrave; 22:30:09

Sujet: Programmation C# .NET

## C# Copier un objet sans implémenter ICloneable

Pour cloner un objet en C#, il est recommand&eacute; d'impl&eacute;menter l'interface ICloneable. On peut ensuite appeler la m&eacute;thode Clone() sur l'objet impl&eacute;mentant la classe ICloneable.

Cependant il n'est pas toujours possible d'impl&eacute;menter l'interface ICloneable si la classe de l'objet &agrave; copier ne nous appartient pas. La plupart des biblioth&egrave;ques externes proposent des m&eacute;thodes Clone quand cela est n&eacute;cessaire mais ce n'est pas toujours le cas.

Lorsque vous n'avez pas acc&egrave;s &agrave; la classe de l'objet &agrave; copier, il reste une solution, copier l'objet en utilisant la serialization binaire.

Ce type de clonage fait une copie bit par bit de l'objet cible. Les r&eacute;f&eacute;rences pointeront donc vers les m&eacute;mes objets que l'objet initial et seuls les types valeurs seront r&eacute;ellement clon&eacute;s.

Ce type de copie ne peut donc pas &ecirc;tre utilis&eacute; dans tous les cas et il faut bien comprendre les impacts avant de l'utiliser.

Voici le code d'un serializer permettant d'effectuer une copie binaire d'un objet m&eacute;me s'il ne poss&egrave;de pas de fonction Clone.

Pour l'utiliser, il vous suffit d'appeler ObjectCloner.Clone(objetACopier).

Exemple:

```
MyClass copie = ObjectCloner.Clone(original);
```

```
using System.IO;
using System.Runtime.Serialization;
using System.Runtime.Serialization.Formatters.Binary;
```

```
public static class ObjectCloner
{
    &nbsp; &nbsp; &nbsp; // &lt;summary&gt;
    &nbsp; &nbsp; &nbsp; // Perform a deep Copy of the object.
    &nbsp; &nbsp; &nbsp; // &lt;/summary&gt;
    &nbsp; &nbsp; &nbsp; // &lt;typeparam name=&quot;T&quot;&gt;The type of object
    &nbsp; &nbsp; &nbsp; // &lt;param name=&quot;source&quot;&gt;The object
    &nbsp; &nbsp; &nbsp; // &lt;/param&gt;
    &nbsp; &nbsp; &nbsp; // &lt;returns&gt;The copied object.&lt;/returns&gt;
    &nbsp; &nbsp; &nbsp; public static T Clone&lt;T&gt;(T source)
    &nbsp; &nbsp; &nbsp; {
    &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; if (!typeof(T).IsSerializable)
    &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; {
```

