

Date: Lundi 02 mai 2011 &agrave; 01:58:03  
Sujet: Programmation C# .NET

## C# Trier un dictionnaire (Dictionary) par valeur

Trier un dictionnaire par valeur peut &ecirc;tre utile en C# si on n'a un dictionnaire de cl&eacute;s et de valeurs. Par exemple : la cl&eacute; peut &ecirc;tre un mot (string) et la valeur un entier (int) repr&eacute;sentant la fr&eacute;quence d'apparition de ce mot dans un texte.

La collection Dictionary peut facilement &ecirc;tre tri&eacute;e par cl&eacute;s mais le tri par valeur est un peu plus compliqu&eacute;.

Il y a plusieurs solutions qui pourrait venir &agrave; l'esprit pour trier un dictionnaire en C#.

Les SortedDictionary sont une collection adapt&eacute;e sauf que le tri se fait sur les cl&eacute;s (key) et non sur les valeurs. C'est toutesfois un type de collection qui peut s'av&eacute;rer fort utile mais dans notre cas, nous ne pourrons pas les utiliser.

Il existe &eacute;galement les SortedList mais celle-ci ne sont utile que pour contenir une liste tri&eacute;e d'un type d'objet (exemple : une liste de mots sous forme de string). Pour trier un dictionnaire en fonction de ses valeurs, nous allons cr&eacute;er une liste de KeyValuePair qui contiendra les 2 &eacute;l&eacute;ments du dictionnaire (cl&eacute; et valeur) tri&eacute;s par valeur.

```
List<KeyValuePair<string, string>> maListe = new List<KeyValuePair<string, string>>(s)
```

```
    maListe.Sort((elt1,elt2) =>
    &nbsp; &nbsp; &nbsp; {
    &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; return elt1.Value.CompareTo(elt2.Value);
    &nbsp; &nbsp; &nbsp; }
    );
```

ou plus simplement :

```
    myList.Sort((x,y)=>x.Value.CompareTo(y.Value));
```

On peut &eacute;galement conserver le dictionnaire initial et utiliser LINQ.

Exemple d'utilisation de LINQ pour trier un dictionnaire par valeur en C# :

```
System.Collections.Generic.Dictionary<string, int> monDictionnaire =
new Dictionary<string, int>();
```

```
&nbsp; &nbsp; myDict.Add(&quot;un&quot;, 1);
&nbsp; &nbsp; myDict.Add(&quot;quatre&quot;, 4);
&nbsp; &nbsp; myDict.Add(&quot;deux&quot;, 2);
&nbsp; &nbsp; myDict.Add(&quot;trois&quot;, 3);
```

```
&nbsp; &nbsp; var DicoTrie = (from entry in monDictionnaire orderby entry.
Value ascending select entry);
```

Bon courage

Publication de Tout sur l'informatic - Programmation C#,  
Scurit, Divx, P2P:  
<http://www.zmaster.fr>

URL de cette publication

<http://www.zmaster.fr/modules.php?name=News&file=article&sid=232>