

Get the names of all the properties of an object

```
#region License and Copyright
```

```
/*
```

```
* Dotnet Commons Reflection
```

```
*
```

```
* Copyright 2005. EDWARD LIM
```

```
*
```

```
* This library is free software; you can redistribute it  
and/or modify it
```

```
* under the terms of the GNU Lesser General Public License as  
published by
```

```
* the Free Software Foundation; either version 2.1 of the  
License, or
```

```
* (at your option) any later version.
```

```
*
```

```
* This library is distributed in the hope that it will be  
useful, but
```

```
* WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY
```

```
* or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser  
General Public License
```

```
* for more details.
```

```
*
```

```
* You should have received a copy of the GNU Lesser General  
Public License
```

```
* along with this library; if not, write to the
```

```
* Free Software Foundation, Inc.,
```

```
* 59 Temple Place,
```

```
* Suite 330,
```

```
* Boston,
```

```
* MA 02111-1307
```

```
* USA
```

```

*
*/
#endregion

using System;
using System.Collections;
using System.Collections.Specialized;
using System.Reflection;

//using Dotnet.Commons.Lang;

namespace Dotnet.Commons.Reflection
{
    ///
    ///

    /// This class contains utility methods that perform
    operations on object properties.
    ///
    ///
    ///
    /// Copyright 2006 by Edward Lim.
    /// All rights reserved.
    ///
    ///
    public sealed class PropertyUtils
    {
    private PropertyUtils() {}

    /// _____
    ///

    /// Get the names of all the properties of an object
    ///
    /// the object to retrieve the properties' names from /// an
    array of property names
    /// _____
    public static string[] GetPropertyNames(object srcObj)
    {

```

```

if (srcObj == null)
throw new System.ArgumentNullException("srcObj");

return GetPropertyNames(srcObj.GetType());

}

/// _____
///
/// Get the names of all the properties of a type
///
/// the type to retrieve the properties' names from /// an
array of property names
/// _____
public static string[] GetPropertyNames(Type objType)
{

if (objType == null)
throw new System.ArgumentNullException("objType");

string[]      propertyNames      =      new
string[objType.GetProperties().Length];

for (int i=0; i< objType.GetProperties().Length; i++) {
propertyNames[i]      =
((PropertyInfo)objType.GetProperties().GetValue(i)).Name; }
return      propertyNames;      }      }      }
[/csharp][[/csharp][[/csharp][[/csharp][[/csharp][[/csharp][[/csharp]
]

```