Set the value of a property that has been declared as an Enum type using reflection

```
#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() {}

/// ————————————————————
///

/// Set the value of a property that has been declared as an Enum type using reflection
///
/// target object in which property is to be set /// name of the property /// value of the object to be set /// ignore case sensitivity /// null if target object or is null, else target
```

```
object with property set if property if found
/// if the propertyName is null or empty
/// ————————————————————
public static object SetEnumTypeProperty(object targetObj,
string propertyName, object propertyValue, bool ignoreCase)
{
if (targetObj == null)
return null;

if ((propertyName == null) || (propertyName.Length < 1)) {
throw new ArgumentException("propertyName cannot be null or
empty."); } if (propertyValue == null) return targetObj;
PropertyInfo                propInfo                =
targetObj.GetType().GetProperty(propertyName); if (propInfo ==
null) return targetObj; if (!propInfo.PropertyType.IsEnum)
throw new ArgumentException("property " + propertyName + " is
not an Enum type"); if (propertyValue is String) propertyValue
= Enum.Parse(propInfo.PropertyType, (string) propertyValue,
ignoreCase);            propInfo.SetValue(targetObj,
Enum.ToObject(propInfo.PropertyType,
Convert.ToUInt64(propertyValue)), null); return targetObj; } }
}
[/csharp][/csharp][/csharp][/csharp][/csharp][/csharp][/csharp
]
```