Get Files / Get Folders methods

```
// crudwork
// Copyright 2004 by Steve T. Pham (http://www.crudwork.com)
//
// This program is free software: you can redistribute it and/or modify
// it under the terms of the GNU General Public License as published by
// the Free Software Foundation, either version 3 of the License, or
// (at your option) any later version.
//
// This program 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 General Public License for more details.
//
// You should have received a copy of the GNU General Public License
// along with This program. If not, see .

using System;
using System.Collections.Generic;
using System.IO;
using System.Text;
using System.CodeDom.Compiler;
using System.Text.RegularExpressions;

namespace crudwork.Utilities
{
```

```
///

/// File Utility
///
public static class FileUtil
{

#region Enums
///

/// File order type for sorting list of filenames
///
public enum FileOrderType
{
///

/// No sort
///
None = 0,

///

/// Sort by filename
///
Filename = 1,

///

/// Sort by extension
///
Extension = 2,

///

/// Sort by the size
///
Size = 3,

///

/// Sort by last modified timestamp
///
```

```
LastWriteTime = 4,

///

/// Sort by file creation timestamp
///
CreationDate = 5,
}
#endregion
#region GetFiles / GetFolders methods
///

/// Shorten the folder name
///
///  ///  ///
public static string[] MakeRelativePath(string[] fileList,
string path)
{
List results = new List();

for (int i = 0; i < fileList.Length; i++) { string file =
fileList[i]; results.Add(file.Replace(path + "", "")); }
return results.ToArray(); } ///
/// sort the file list based on the given order type
///

///  ///  ///
public static string[] OrderFileBy(string[] fileList,
FileOrderType fileOrderType)
{
string[] orderKey = new string[fileList.Length];
string[] orderVal = new string[fileList.Length];

//int maskLength = StringUtil.MaxLength(fileList);
int maskLength = 100;
string maskFormat = String.Format(@"{{0,{0}}}", maskLength);

for (int i = 0; i < fileList.Length; i++) { string filename =
fileList[i]; string orderByKey; if (!File.Exists(filename))
```

```
throw new FileNotFoundException(filename); FileInfo fi = new FileInfo(filename); switch (fileOrderType) { case FileOrderType.None: orderByKey = ""; break; case FileOrderType.Filename: { orderByKey = String.Format(maskFormat, fi.Name); } break; case FileOrderType.LastWriteTime: { DateTime dt = fi.LastWriteTime; orderByKey = String.Format("{0:0000}{1:00}{2:00}{3:00}{4:00}{5:00}{6:000}", dt.Year, dt.Month, dt.Day, dt.Hour, dt.Minute, dt.Second, dt.Millisecond ); } break; default: throw new ArgumentOutOfRangeException("not supported: " + fileOrderType); } orderKey[i] = orderByKey; orderVal[i] = fileList[i]; } if (fileOrderType != FileOrderType.None) { Array.Sort(orderKey, orderVal); } return orderVal; } #endregion } } [/csharp]
```