Data Juggler Ultimate Helper User's Guide

By Data Juggler Software http://www.datajuggler.com

Description

This class is an attempt to reduce the number of classes I copy to multiple projects. I end up with out of date copies of code and mismatched versions between my home and work computers.

I invite other developers to share their favorite C# helper classes.

It is not an actual requirement that the helper class implements static methods, but static methods are preferred over instance classes.

It is required that the class name end in Helper.

Data Juggler Ultimate Helper Classes

BooleanHelper

The BooleanHelper is designed to make working with Boolean values easier. This method returns the Boolean value for the source string given.

Boolean Helper Methods

ParseBoolean(string sourceString, bool defaultValue, bool errorValue)

This method returns the Boolean value for the source string given.

```
public static bool ParseBoolean(string sourceString, bool defaultValue, bool
errorValue)
```

ConfigurationHelper

This class is used to read from the Web.config or App.config files.

Configuration Helper Methods

ReadAppSetting(string settingName, bool dataIsEncrypted = false, string encryptionKey = "")

This method is used to read values from the Web.config or app.config files.

There are two useful utilities that work in conjunction with the ConfigurationHelper. Dowload RAD Studio Code Generation Toolkit from http://radstudio.codeplex.com . In the Tools folder of the RAD Studio download you will find two projects that are useful:

RAD Studio Configuration Manager
RAD Studio Cryptography Helper

```
public static string ReadAppSetting(string settingName, bool dataIsEncrypted = false, string encryptionKey = "")
```

CrypotgraphyHelper

This class is used to encrypt and decrypt strings.

As mentioned above the RAD Studio download contains a Windows Form sample using the CryptographyHelper class.

CrypotgraphyHelper Methods

`EncryptString(string stringToEncrypt, string password)`

This method is used to encrypt the string given using the ECB (Electronic Code Book Cypher). The string can only be decrypted by the string given.

There is also an override for this method that uses the default password of pokerpro.

```
public static string EncryptString(string stringToEncrypt, string password)
```

`----`

DecryptString(string stringToDecrypt, string password)

This method is used to decrypt the string given. The password used must be the same that was to encrypt the string.

There is an override for this method that uses the default password of pokerpro.

```
public static string DecryptString(string  stringToDecrypt, string password)
```

Date Helper

This class is used to parse and handle date functions.

Date Helper Methods

GetShortDateTime(DateTime date)

This method returns the Date and Time for the date given.

```
public static string GetShortDateTime(DateTime date)
```

IsAfter(DateTime sourceDate, DateTime targetDate, bool includeTime = true)

This method returns true if the TargetDate comes AFTER the sourceDate.

```
public static bool IsAfter(DateTime sourceDate, DateTime targetDate, bool
includeTime = true)
```

IsBefore(DateTime sourceDate, DateTime targetDate, bool includeTime = true)

This method returns true if the TargetDate comes BEFORE the sourceDate.

```
public static bool IsBefore(DateTime sourceDate, DateTime targetDate, bool
includeTime = true)
```

IsDateInRange(DateTime sourceDate, int allowedDays)

This method returns true if the sourceDate is not any older than the allowed days. This is useful if you want to know if a date is older than 30 days old for example.

```
public static bool IsDateInRange(DateTime sourceDate, int allowedDays)
```

IsSameDay(DateTime sourceDate, DateTime targetDate)

This method returns true if the targetDate is the same date as the sourceDate. Time is not a factor in this method as dates are compared by month, day and year only.

```
public static bool IsSameDay(DateTime sourceDate, DateTime targetDate)
```

ParseDate(string sourceString, DateTime defaultValue, DateTime errorValue)

This method is used to safely parse a string. If the sourceString does not exist, the defaultValue is returned. If an error occurs the errorValue is returned.

```
public static DateTime ParseDate(string sourceString, DateTime defaultValue,
DateTime errorValue)
```

List Helper

This class contains helper methods for dealing with lists and collections.

This is one of my favorite classes and the reason I created this project.

RAD Studio was in need of an update to work the Generic Lists as the predecessor to RAD Studio (Data Class Builder.Net) was developed prior to Generic Lists being introduced in the Visual Studio 2005 release I believe.

ListHelper only contains two methods but in my opinion these are two of the most useful methods I have created (borrowed, found, or stole) in my 15 years as a developer (11 for C#).

List Helper Methods

ConvertToListOf<T>(IList sourceList)

This method converts an IList to a Generic List.

RAD Studio originally returned stronged typed collections; collections of type System.Collections.CollectionBase. I have grown very fond of using generic lists and have never had the time to research how to add an option for RAD Studio to return generic lists. After I created this method I have never had a priority reason to modify RAD Studio.

If it works, works really fast and works really well, why fix it?

```
public static object ConvertToListOf<T>(IList sourceList)
```

HasOneOrMoreItems(IList list)

This method returns true if the List specified exists and hast 1 or more items.

I bet you wish you had a dollar for every time you have typed something like this:

```
// if there are one or more items
if ((list != null) && (list.Count > 0))
{
   // do something …
}
```

You can pass in any collection or array that derives from IList, and most can.

```
public static bool HasOneOrMoreItems(IList list)
```

This example (borrowed from Count Widget – www.countwidget.com or http://countwidget.codeplex.com) uses both the ConvertToListOf method and the HasOneOrMoreItems methods:

```
// if there are one or more addresses
if (ListHelper.HasOneOrMoreItems(tempCollection))
{
   // set the return value
   users = (List<User>) ListHelper.ConvertToListOf<User>(tempCollection);
}
```

Message Box Helper

This class is used in Windows Forms applications and WPF applications to show a message to the user or to get a users response.

Message Box Helper Methods

GetUserResponse(string message, string title, MessageBoxButtons buttons = MessageBoxButtons.YesNo, MessageBoxIcon icon = MessageBoxIcon.Question)

This method returns the users response (DialogResult) that a user gives to a given message. This is used to get confirmation for delete or asking the user for permission to perform an action, etc.

```
public static GetUserResponse(string message, string title, MessageBoxButtons
buttons = MessageBoxButtons.YesNo, MessageBoxIcon icon =
MessageBoxIcon.Question)
```

ShowMessage(string message, string title, MessageBoxButtons buttons = MessageBoxButtons.OK, MessageBoxIcon icon = MessageBoxIcon.Warning)

The ShowMessage method is used to make it easy to show a message to the user. The Visual Studio message box has so many overrides that is difficult to work with.

This method covers the basics for showing a message:

1. Setting the Title for the message
2. Setting the message text
3. Optionally you can pass in the buttons to show
4. And finally also there is an option to change the icon.

These two methods cover 95% of my MessageBox uses if not higher.

```
public static void ShowMessage(string message, string title, MessageBoxButtons
buttons = MessageBoxButtons.OK, MessageBoxIcon icon =
MessageBoxIcon.Warning)
```

MouseHelper

MouseHelper is used to click the mouse in Windows and WPF applications.

MouseHelper requires a reference to System.Drawing and I was on the fence as to whether MouseHelper is generic enough to be included in an all purpose helper solution. I finally decided to included MouseHelper because System.Drawing does not affect web servers and deployment is not an issue since System.Drawing is in included in the .Net Framework.

You can always remove MouseHelper if including a reference to System.Drawing is not practical in your particular architecture.

MouseHelper Methods

MouseClick(Point point)

This method moves the cursor to the point specified and simulates a mouse click.

```
public static void MouseClick(Point point)
```

There is also an override that accepts integer x and y values:

```
public static void MouseClick(int x, int y)
```

NullHelper

The NullHelper class is used to test for nulls with the IsNull method and to test for not equal to null with the Exists method.

This class is especially useful for compound if statements as there are overrides for one to four objects can be tested for each method.

NullHelper Methods

Exists(object item, object item2, object item3, object item4)

This method tests if the objects passed in exist.

You can pass in one to four items with this method.

```
public static bool Exists(object item, object item2, object item3, object item4)
```

IsNull(object item, object item2, object item3, object item4)

This method is used to test if any of the objects passed in are null. In contrast to the Exists method which uses ‘and’ this method uses ‘or’ to test for null.

You can pass in one to four items with this method.

```
public static bool IsNull(object item, object item2, object item3, object item4)
```

NumericHelper

This class contains helper methods for working with integers, doubles, prime numbers, rounding and more.

If you have any favorite methods that make working with numbers easier please send them to support@datajuggler.com

NumericHelper Methods

GetNumericValueOfWord(string word)

This method returns the numeric value of the word given.

This is useful in cryptography or can be used a check digit in routines that need an extra layer of security.

```
public static int GetNumericValueOfWord(string word)
```

GetPrimeNumberAfter(int startAfter)

This method returns the prime number after the seed given.

This is useful in cryptography or other routines to find the first prime number after the startAfter value given.

```
public static int GetPrimeNumberAfter(int startAfter)
```

IsNumberPrime(int number)

This method returns true if the Number given is a prime number.

public static bool IsNumberPrime(int number)

ParseDouble(string sourceString, double defaultValue, double errorValue)

This method is used to safely parse a string into a double.

If the sourceString given does not exist, the defaultValue is returned. If an error occurs parsing the string the errorValue is returned.

public static double ParseDouble(string sourceString, double defaultValue, double errorValue)

ParseInteger(string sourceString, int defaultValue, int errorValue)

This method is used to safely parse a string into an integer.

If the sourceString given does not exist, the defaultValue is returned. If an error occurs parsing the string the errorValue is returned.

public static int ParseInteger(string sourceString, int defaultValue, int errorValue)

ReturnDecimalRemainderAsDouble(double sourceDouble, double defaultValue, double errorValue)

This method returns the Decimal Remainder As a Double
Example: 23.45 would return .45

public static double ReturnDecimalRemainderAsDouble(double sourceDouble, double defaultValue, double errorValue)

ReturnDecimalRemainderAsInteger(double sourceDouble, int defaultValue, int errorValue)

This method returns the Decimal Remainder cast as an integer.

public static int ReturnDecimalRemainderAsInteger(double sourceDouble, int defaultValue, int errorValue)

RoundDown(double sourceDouble)

This method returns the sourceDouble minus the remainder.

public static int RoundDown(double sourceDouble)

RoundUp (double sourceDouble)

This method returns the sourceDouble minus the remainder + 1.

public static int RoundDown(double sourceDouble)

ParentControlHelper

This class is used to find the parent Web Control for a source Web User Control given.

This class is used only for web development projects to find the parent control for the source control given.

There are overrides for this method to search by ControlID or by the Type of the parent object.

ParentControlHelper Methods

GetParentUserControl(UserControl sourceControl, string controlID, int maxAttempts = 0)

This method searches recursively until it finds the parent control for the source control given. This method will loop until the maxAttempts has reached if maxAttempts is > 0, else it will continue until a parent control is not found.

public static UserControl GetParentUserControl(UserControl sourceControl, string controlID, int maxAttempts = 0)

Or

public static UserControl GetParentUserControl(UserControl sourceControl, Type parentType, int maxAttempts = 0)

PluralWordHelper

This class is used to return the plural word for the source word given.

A Dictionairy approach would be a more extensible solution, but I am using a switch statement for ease of development.

I have only stubbed out a few words as a starter. This was created for RAD Studio but I have not been very diligent about updating this class.

Please send me any updates to support@datajuggler.com and I will include them in the next release.

PluralWordHelper Methods

GetPluralName(string singularWord, bool returnLowerCase)

This method is used to get the plural word for the singularWord given.

An example of this is Property returns Properties, not Propertys.

If the word is not handled than an s is added to the end of the word.

Optionally you can specify returnLowerCase and the word will make a ToLower() call before returning.

```
public static string GetPluralName(string singularWord, bool returnLowerCase)
```

SqlParameterHelper

This class is used to create the Sql parameters for stored procedures.

This is used in conjunction with RAD Studio Code Generation Toolkit (http://radstudio.codeplex.com) but you can use it without RAD Studio.

SqlParameterHelper Methods

CreateSqlParameters(string parameterName, object parameterValue)

You can pass in one to four parameter name / value pairs.

```
public static SqlParameter[] CreateSqlParameters(string parameterName, object
parameterValue)
```

SystemHelper

This class is used to get information abou the current system and executing application name and Guid.

If you have any favorite System methods please send an email to support@datajugler.com.

More methods will be added to this class as needed.

SystemHelper Methods

GetSystemProductGuid(AssemblyTypeEnum assemblyType = AssemblyTypeEnum.EntryAssembly)

This method returns the 'Entry' assembly.Guid or the 'Executing' assembly.Guid. The Entry' assembly is the assembly that executed that caused the ExecutingAssembly to run. These may or not be the same assembly depending on if this is called by the main application assembly or by a .dll that is referenced by the main application assembly.

<param name="assemblyType">Specify if you want the Guid from the 'Entry' assembly or the 'Executing' assembly.</param>

```
public static string GetSystemProductGuid(AssemblyTypeEnum assemblyType =
AssemblyTypeEnum.EntryAssembly)
```

GetSystemProductName(AssemblyTypeEnum assemblyType = AssemblyTypeEnum.EntryAssembly)

This method returns the 'Entry' assembly.Name or the 'Executing' assembly name. The EntryAssembly is the assembly that executed that caused the Executing assembly to run. These may or not be the same assembly depending on if this is called by the main application assembly or by a .dll that is referenced by the main application assembly.

<param name="assemblyType">Specify if you want the name from the 'Entry' assembly or the 'Executing' assembly.</param>

```
public static string GetSystemProductName(AssemblyTypeEnum assemblyType =
AssemblyTypeEnum.EntryAssembly)
```

TextHelper

This class contains helper methods for dealing with strings.

TextHelper Methods

CombineStrings(string string1, string string2)

This method appends string2 to the end of string1.

There are overrides that allow you to append one to three strings to the end of string1.

```
public static string CombineStrings(string string1, string string2)
```

CombineStringsEx(string string1, string string2, string separator="")

This method which is based upon the CombineString methods appends string2 to the end of string1. This method allows you to pass in a separator string such as a comma, space or dash to create delimeted strings.

If you do not pass in a separator string the method behaves exactly as the CombineStrings method.

There are overrides that allow you to append one to three strings to the end of string1. If a separator is passed in it will be inserted between each of the string parameters (string1, string2, string3 or string4).

```
public static string CombineStrings(string string1, string string2, string separator =
"")
```

Exists(string sourceString)

This method is used to test if one to four strings exists.

If you call one of the overrides with two or more strings, all of the strings must exist or the method will return false.

Personally I have always hated String.IsNullOrEmpty – Exists is much shorter.

```
public static bool Exists(string sourceString)
```

CapitalizeFirstChar(string word, bool lowerCase = false)

This method capitalizes the first character of a word.
Optionally you can pass in lowercase = true and the first character will be made lowercase.

```
public static string CapitalizeFirstChar(string word, bool lowerCase = false)
```

GetTextLines(string sourceText)

This method returns a list of TextLine objects.
The strings are parsed you the Environment.Newline characters so it is possible some text lines will not be parseable.

You will need a reference to UltimateHelper.Objects to reference the TextLine object. Also you will need a reference to System.Collections.Generic to reference the List<TextLine> that is returned from this method.

This method was created for Regionizer (http:/Regionizer.codeplex.com). Once you download any Data Juggler Software code you will want to install Regionizer as that is how all of our code is formatted into regions.

```
public static List<TextLine> GetTextLines(string sourceText)
```

GetWords(string sourceText)

This method returns all of the words in a string.

You will need a reference to UltimateHelper.Objects to reference the Word object.

The delimeters are hardcoded for now, you can modify the list of delimeters if you need something different.

String.Split is used to separate the strings into words.

```
public static List<Word> GetWords(string sourceText)
```

IsEqual(string sourceString, string sourceString2, bool textMustExist = true, bool caseMustMatch = false)

This method returns true if the two strings given are equal.

<param name="sourceString">The source string</param>
<param name="sourceString2">The string to be compared to.</param>
<param name="textMustExist">This defaults to true; if true both strings must exist or false is returned even if the strings match.</param>
<param name="caseMustMatch">This defaults to false; if true the comparison will be a case sensitive comparison.

```
public static bool IsEqual(string sourceString, string sourceString2, bool
textMustExist = true, bool caseMustMatch = false)
```

Final Notes

I do not claim that this code is error free. If you find any errors, suggestions, a better way of doing things or to submit your own classes or a kind word please send an email to support@datajuggler.com .

If you like please rate this project and review it as in 3 ½ years and nearly 10,000 combined downloads out of 10 projects and I have not received a single rating.

Even a one star rating would convince me that all of the downloads were not bots.

If the Matrix is real and I am inventing all of this then why do I invent that other projects get ratings and reviews? I might ask such questions because I took the red pill, or was it the blue pill, or does that count as purple if I took both?

I have tested and used all of this code in this project at some point or another. A few methods were added and cleaned up as I wrote the documentation so I guess it is always important to document.

If you like this project or any of the Data Juggler projects, please support my website, www.daily-click.com, which is dedicated to using technology to helping people and animals.

Please visit http://www.datajuggler.com to see other great products from Data Juggler Software