Objects Comparer

# Introduction

Objects comparer is object-to-object comparer, which allows you recursively compare objects member by member and define comparison rules for certain properties, fields or types.

Objects comparer can be considered as ready to use framework or as an idea for similar solutions.

# Installation

# Install-Package ObjectsComparer

# Basic Example

public class ClassA

{

public string StringProperty { get; set; }

public int IntProperty { get; set; }

# }

var a1 = new ClassA { StringProperty = "String", IntProperty = 1 };

var a2 = new ClassA { StringProperty = "String", IntProperty = 1 };

var comparer = new Comparer<ClassA>();

var isEqual = comparer.Compare(a1, a2);

Debug.WriteLine("a1 and a2 are " + (isEqual ? "equal" : "not equal"));

a1 and a2 are equal

var a1 = new ClassA { StringProperty = "String", IntProperty = 1 };

var a2 = new ClassA { StringProperty = "String", IntProperty = 2 };

var comparer = new Comparer<ClassA>();

IEnumerable<Difference> differenses;

var isEqual = comparer.Compare(a1, a2, out differenses);

var differensesList = differenses.ToList();

Debug.WriteLine("a1 and a2 are " + (isEqual ? "equal" : "not equal"));

if (!isEqual)

{

Debug.WriteLine("Differences:");

Debug.WriteLine(string.Join(Environment.NewLine, differensesList));

}

a1 and a2 are not equal

Differences:

Difference: MemberPath='IntProperty', Value1='1', Value2='2'

# Overriding comparison rules

# Utility Classes

## Factory

## Non-generic comparer

## Predefined Value Comparers

# Examples

## Example 1: Expected Message

## Example 2: Persons comparison