**GetMonthRangeUTC Challenge**

Abdessattar Hayouni

[Abdessattar.hayouni@ucalgary.ca](mailto:Abdessattar.hayouni@ucalgary.ca)

<http://hayouni.com>

1. Discussion

void GetMonthRangeInUtc(DateTime aDate, out DateTime utcMonthStart, out DateTime utcNextMonthStart)

{

// compute the first day of the month containing aDate and successive months

DateTime[] monthStart = new DateTime[2];

for (int i = 0; i <= monthStart.Length; i++)

{

monthStart[i] = new DateTime(aDate.Year, aDate.Month++, 1);

}

// Compute the offset from UTC to our local time (UTC + offset = localtime).

TimeSpan utcOffset = TimeZone.CurrentTimeZone.GetUtcOffset(aDate);

// convert local times to UTC (UTC = localtime - offset)

utcMonthStart = monthStart[0].Subtract(utcOffset);

utcNextMonthStart = monthStart[1].Subtract(utcOffset);

}

The following bugs and thoughts are noted concerning the implementation above:

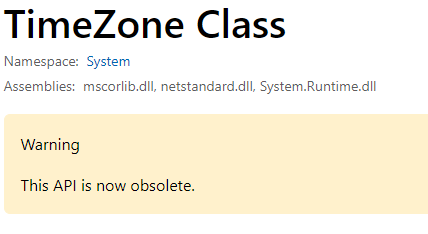
* By using the ++ operator the compiler will attempt to update the value of aDate.Month which is read-only, resulting in compilation error. Instead we should use +1 which takes the value already assigned to aDate.Month, adds one to it and assign it to the new DateTime instance monthStart[i] without attempting to reassign aDate.Month.
* A correct implementation will lead to the following array monthStart:

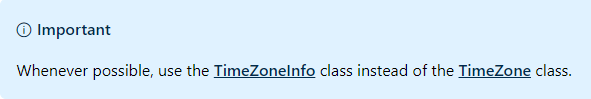
|  |  |  |
| --- | --- | --- |
| Index | 0 | 1 |
| Value | aDate.Month | aDate.Month+1 |

Therefore instead of adding 1 to aDate.Month , we should add the counter value (i) as following:

monthStart[i] = new DateTime(aDate.Year, aDate.Month+i, 1);

* TimeZone is deprecated:





Src: <https://docs.microsoft.com/en-us/dotnet/api/system.timezone?view=netframework-4.7.2>

Hence , we will use TimeZoneInfo class instead .

* The choice of variable names should be self explanatory and not misleading, the variable monthStart has the current month at index 0 and the next month at index 1, therefore it is better to give it a more relevant name such as monthBoundaries or monthEdges …

**2-Solution**

I present the solution below after fixing the bugs and making the changes noted above:

using System;

namespace CS

{

class Program

{

static void Main(string[] args)

{

*DateTime* utcMonthStart;

*DateTime* utcNextMonthStart;

*DateTime* aDate = new *DateTime*(2014, 2, 1);

GetMonthRangeInUtc(aDate, out utcMonthStart, out utcNextMonthStart);

//Your code goes here

Console.WriteLine("The first second of the first day of the month is:{0}(UTC)",utcMonthStart);

Console.WriteLine("The first second of the first day of the next month: {0} (UTC)",utcNextMonthStart);

}

static void GetMonthRangeInUtc(*DateTime* aDate, out *DateTime* utcMonthStart, out *DateTime* utcNextMonthStart)

{

// compute the first day of the month containing aDate and successive months

*DateTime*[] monthBoundaries = new *DateTime*[2];

for (int i = 0; i < monthBoundaries.Length; i++)

{

monthBoundaries[i] = new *DateTime*(aDate.Year, aDate.Month+1, 1);

}

// Compute the offset from UTC to our local time (UTC + offset = localtime).

*TimeSpan* utcOffset = TimeZoneInfo.Local.GetUtcOffset(aDate); //use TimeZoneInfo instead of TimeZone

*TimeZoneInfo* localZone = TimeZoneInfo.Local;

Console.WriteLine("Local Time zone:{0} , UTC Offset: {1}", localZone.StandardName,utcOffset);

// convert local times to UTC (UTC = localtime - offset)

utcMonthStart = monthBoundaries[0].Subtract(utcOffset);

utcNextMonthStart = monthBoundaries[1].Subtract(utcOffset);

}

}

}

1. **Results**



1. **Reference**

Github: <https://github.com/hayouni15/GetonthRangeInUtc>