

Re: Converting the time from one timezone to another

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework/2006-02/msg00625.html>

- *From:* "Nathan Sokalski" <njsokalski@xxxxxxxxxxxx>
 - *Date:* Sat, 18 Feb 2006 17:31:44 -0500
-

Did you read my posting? I understand the concept of UTC, and the process of getting from servertime to localtime (I even showed the code for this in my posting). My question is "Does .NET have a way of telling me how many hours to subtract, perhaps a function that returns the number of hours to add/subtract from UTC that might be something like this:

UtcOffsetForTimezone("EST") would return -5

If I wanted to get the local time for Eastern Standard Time, I could very easily just use

```
Date.UtcNow.AddHours(-5)
```

The reason I don't like this method is because since most people know what timezone they live in by it's name rather than it's UTC offset, I would like a way to enter the timezone as a parameter in order to find the offset. Sure, I could lookup all the timezones and their offsets myself, but I would expect this to already be a part of ASP.NET. This may sound lazy, but if you look at all the other things that are easy but ASP.NET have done for you, it's not an unreasonable thing to ask.

—

Nathan Sokalski
njsokalski@xxxxxxxxxxxx
<http://www.nathansokalski.com/>

"Stephany Young" <noone@localhost> wrote in message
<news:OVjtB6MNGHA.1716@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

Given the other newsgroups you cross-posted to, I have to assume that you are talking about an ASP.NET application running on a web server halfway around the world, and that your reference to 'local' actually refers to the machine upon which you are viewing the generated 'page' in a web browser.

Working on that principle, I also assume that you want to either:

Re: Converting the time from one timezone to another

1. Carry out some calculation at the server using a date/time passed from the client that represents a point in time at the client expressed in terms of the time zone where the server is physically situated,

or

2. Carry out some calculation at the client using a date/time passed from the server that represents a point of time at the server expressed in terms of the time zone where the client is physically situated.

Either way, the most critical aspects of the whole exercise are that:

1. The time zone for both the 'server' and 'client' machines are set correctly,

and

2. The date and time for both the 'server' and 'client' machines are regularly corrected using a reliable source.

The key to solving the 'problem' is to forget about time zones per se, and to think in terms of a common point of comparison and the differences from that point.

In terms of date/time we, happily, have the concept of Universal Time Coordinated (UTC) which provides a common reference point for expressing date/time anywhere in the world. For those who haven't caught up yet, UTC used to be known as Greenwich Mean Time (GMT).

We know that your 'client' machine is physically situated somewhere within the Eastern Time (US & Canada) time zone which is 5 hours behind UTC. We will take you literally and assume that your 'server' machine is physically located 'halfway around the world', in a time zone that is 5 hours ahead of UTC.

As long as both machines are configured correctly, they will be aware of daylight saving factors relating to their respective time zones and, therefore daylight saving adjustments will be applied to any date/time calculations automatically.

For 'one end' to be able to express date/time factors in terms of the 'other end' then one end has to:

1. Know in advance where the other end is, in relation to UTC,

or

2. Be told as required what the date/time is at the other end, in terms

Re: Converting the time from one timezone to another
of UTC.

Knowing in advance where the other end is, in relation to UTC, is not really practical for an ASP.NET application because it would need to know in advance the location of every machine that could ever submit a request of it.

So that leaves us with communicating the necessary information as and when required.

The DateTime structure nicely presents us with properties and methods for dealing with values expressed in terms of UTC.

To 'capture' the local date and time as UTC we use:

```
Dim _utc As DateTime = DateTime.UtcNow
```

```
MySendUTCToOtherEnd(_utc)
```

To convert a UTC date and time to local we use:

```
Dim _remoteutc as DateTime = MyGetUTCFromOtherEnd()
```

```
Dim _local As DateTime = _remoteutc.ToLocalTime
```

_local now contains a value that represents the date and time from the remote machine expressed in terms of the time zone where the local machine is physically located.

To illustrate with an example:

Client machine is UTC - 5 hours
Server machine is UTC + 5 hours

Client local date/time = February 18, 7:00 PM
Server local date/time = February 19, 5:00 AM

UTC value passed from client = February 19, 0:00 AM

_local as calculated by server = February 19, 5:00 AM

and, in reverse:

Client machine is UTC - 5 hours
Server machine is UTC + 5 hours

Server local date/time = February 19, 5:00 AM
Client local date/time = February 18, 7:00 PM

UTC value passed from server = February 19, 0:00 AM

Re: Converting the time from one timezone to another

_local as calculated by client = February 18, 7:00 PM

"Nathan Sokalski" <njsokalski@xxxxxxxxxxxx> wrote in message
news:OZBJ1uFNGHA.2300@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

I asked a question about a week ago about how to get my local time, since my application is running on a server halfway around the world. I have determined that my code would look something like this:

```
Dim servertime As Date = Date.Now
```

```
Dim utctime As Date = servertime.ToUniversalTime()
```

```
Dim localtime As Date
```

I know that the last step would be to adjust the utctime value by the appropriate amount using code such as `utctime.AddHours(-5)`, but I am looking for a way to get this value by supplying the timezone rather than an offset (in other words, I am looking for a function that returns either a `System.TimeSpan` or `Integer` when I enter the timezone) so that I can do something such as

```
utctime.AddHours(GetTZOffset(TimeZones.EST))
```

Is there a function that does this, or any way to get the offset by submitting the timezone? Thanks.

--

Nathan Sokalski
njsokalski@xxxxxxxxxxxx
<http://www.nathansokalski.com/>