

## Re: adding time segments

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2007-10/msg01319.html>

---

- *From:* Peter Duniho <[NpOeStPeAdM@xxxxxxxxxxxxxxxxxxxx](mailto:NpOeStPeAdM@xxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Tue, 09 Oct 2007 11:38:48 -0700
- 

Paul wrote:

[...]

I currently have a loop and am comparing the current entry with the previous entries. This works ok until I get to the 3:00–3:30 segment as it does not know to not count it since it is only comparing it with the 6:30–7:00 segment.

Can we assume that all of the times being considered are either on the same date, or include the date information?

It seems to me that the most straightforward implementation would be to keep some kind of list of "DateTime with range" instances. In other words, instances of a class that looks something like this:

```
class DateTimeRange
{
    DateTime _dtStart;
    DateTime _dtEnd;

    public DateTimeRange(DateTime dtStart, DateTime dtEnd)
    {
        _dtStart = dtStart;
        _dtEnd = dtEnd;
    }
}
```

You'd start with an empty list. Each time you add a new time range (eg "starting at 3pm, ending at 4pm", you would scan the list, looking for a place to insert it. If there's overlap, you would modify the existing entry in your list, expanding it to include the new entry. You do this for each entry in your list of time ranges. When you're done, you simply enumerate the list and extract the final start/end time values from the list.

You'll have a variety of cases to deal with. The overlap could involve the new range ending within an existing range, beginning within an existing range, being completely contained within an existing range, completely containing an existing range, or even spanning two existing ranges.

I think a general-purpose way of dealing with those cases would be to implement the code so that you simply check for any kind of overlap, and once you find overlap, you remove the existing range from the list, merge it as needed with the new range, and then re-add the result to your list, handling the newly created range just

Re: adding time segments

as you would an original.

Conceptually I think that implementing that recursively would be easiest, but it's not actually required.

Hope that helps.

Pete

.