

NET: Time Zone

[Stampa](#)

[\(10\)](#) » [NET: Installazione Web Services Enhancements \(WSE\) 3.0 per VS 2005](#) » [NET: SubVersion](#) » **[NET: Time Zone](#)**

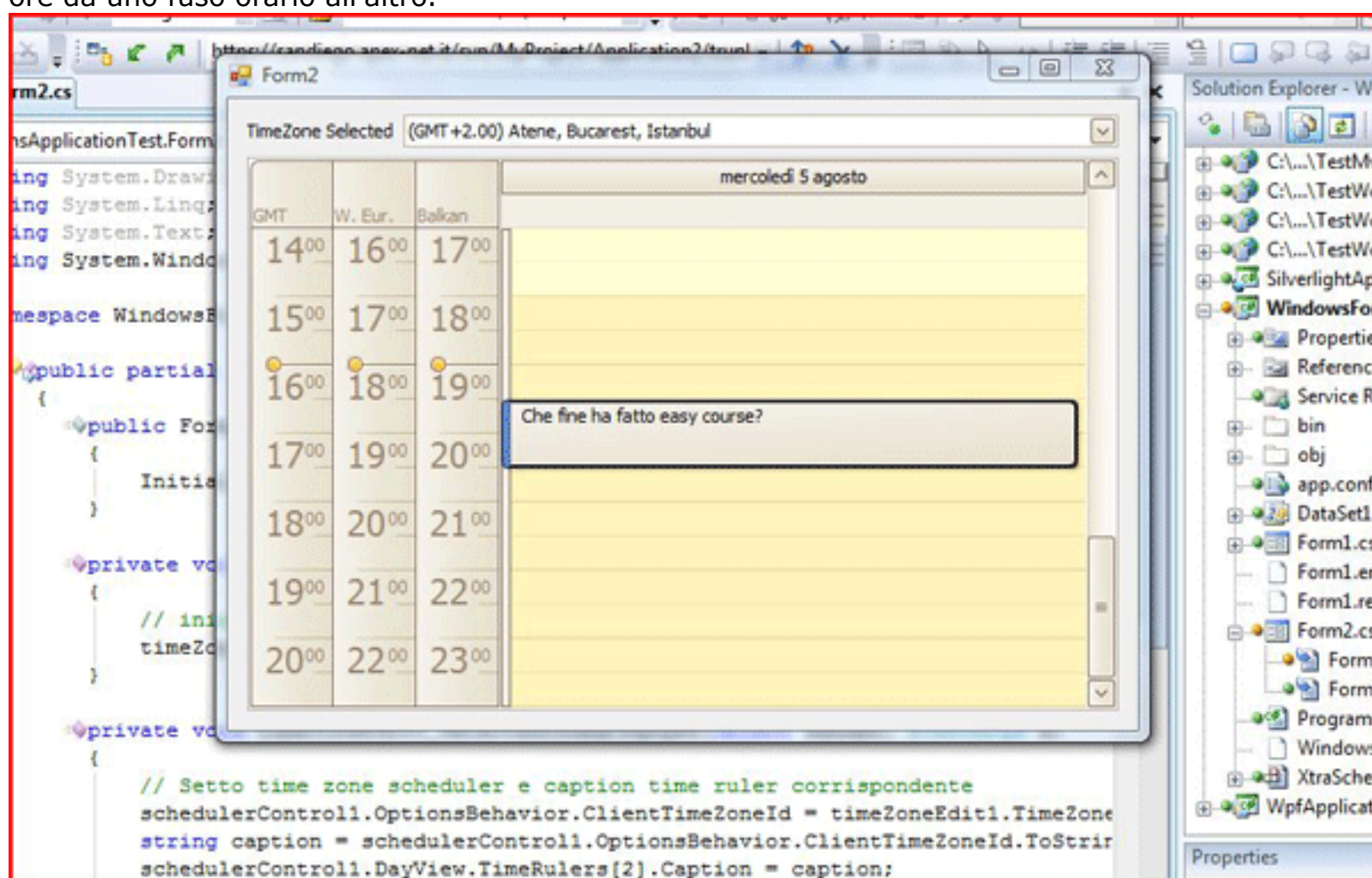
Indice dei Contenuti [[Nascondi/Mostra](#)]

[Scenario: Time Zone](#)

[Riferimenti](#)

Scenario: Time Zone

- Quando si sviluppano applicazioni internazionali, è a volte necessario utilizzare i fusi orari in modo corretto; occorre ad esempio creare applicazioni dipendenti dal fuso orario in grado di convertire date e ore da uno fuso orario all'altro.



Management Time Zone Example

- Potresti dover schedulare ad esempio un appuntamento per una call-conference tra utenti di **time-zone differenti**, l'appuntamento **unico** dovrà essere **visto** secondo il proprio fuso orario e con l'applicazione della eventuale ora legale
- In alcuni casi particolari potresti dover gestire i cosiddetti "time zone independent **floating** appointments": se i miei uffici sono localizzati in differenti paesi nel mondo, potresti disporre che il mio staff cominci a lavorare alle 9 AM; significa che non cominceranno a lavorare simultaneamente, ma in ogni zona il tempo locale di inizio lavori sarà lo stesso; la gestione di un caso come questo è meno banale di quello che potrebbe sembrare
- Quale che sia il concetto applicativo che voglio gestire connesso al fuso orario, occorre avere chiaro cosa viene gestito dal proprio stack applicativo e come posso interagire con esso
=== Piattaforma .NET ===
- Oltre alla struttura DateTime di base, .NET Framework include le classi seguenti che supportano l'utilizzo dei fusi orari:
 - TimeZone
 - TimeZoneInfo**: semplifica la creazione di applicazioni che dipendono dal fuso orario
 - DateTimeOffset

```
1 // c# exampleDateTime dateNow = DateTime.Now; DateTime dateUTC = TimeZoneInfo.Conve?
```

- Al fine di prevenire possibili discrepanze sarebbe bene memorizzare le informazioni di data e ora in formato **UTC** (Coordinated Universal Time); anche se a stretto rigore non sarebbe necessario
- Vi sono dei meccanismi **impliciti** che fanno le conversioni nelle varie time-zone, ad esempio nella comunicazione tra un client e un web service entrambi .NET, di cui occorre essere coscienti se eventualmente si vuole andare in override
- **Devexpress**: nel caso dei controlli di tipo **Scheduler**, sia windows che web, è prevista una completa gestione del time zone sia a livello di presentation (agendo per esempio sulla proprietà SchedulerOptionsBehaviorBase.ClientTimeZoneId) che dal punto di vista degli storage (agendo per esempio sulla proprietà AppointmentStorageBase.DateSaving); sono presenti inoltre dei particolari controlli per l'interazione della UI (time zone edit) o classi di utilità come TimeZoneHelper

Riferimenti

- [MSDN: Times and Time Zones](#)
- [Override Conversioni Locali per Datetime in Web Services](#)
- [Devexpress On line Doc.: XtraScheduler Time Zone](#)