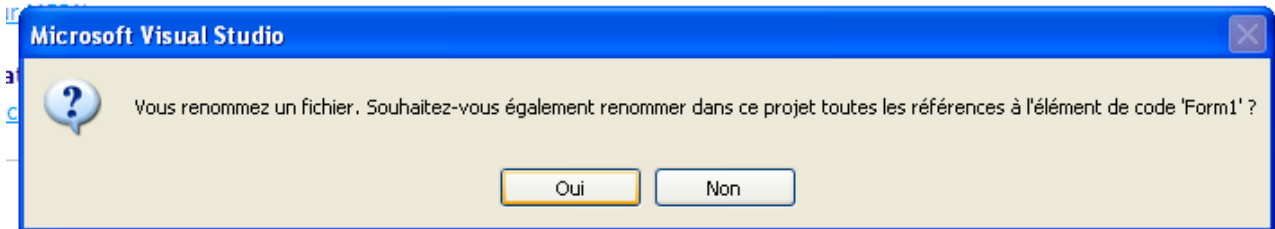


TP Timer

I) Création du projet

- Créer un nouveau projet C#, Application Windows forms intitulé: **ExempleTimer**
- Dans l'explorateur de solutions, renommer (clic droit) Form1.cs en Formtimer.
- A l'affichage de la boîte de dialogue suivante, répondre Oui.



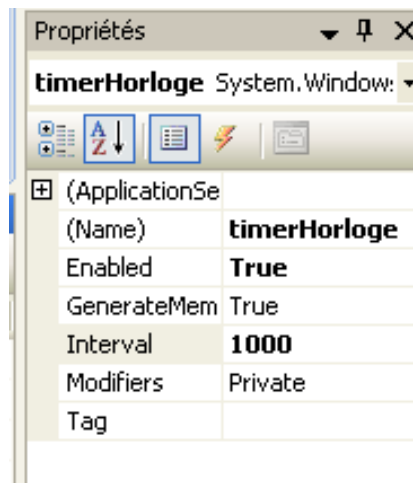
- Définir la propriété Text du formulaire à: **Exemple de timer**

II) Mise en place du Timer.

- Ajouter à partir de la boîte à outils (catégorie composants) un contrôle Timer (double clic).



- Définir les propriétés du contrôle Timer comme indiqué ci-dessous:

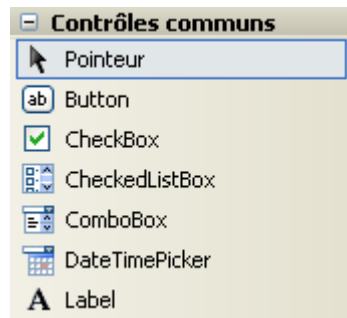


- La propriété Interval définit l'intervalle de temps entre chaque déclenchement de l'événement Tick. La propriété Interval est définie en ms.

Le système d'exploitation compare en permanence son horloge interne à l'intervalle défini par la propriété Interval, et donne l'ordre au Timer d'exécuter son événement Tick. Attention, les timers sont gros consommateurs de ressources système!

- La propriété enabled est fixée à True, ce qui active le timer.

III) Création d'un label qui affichera l'heure système grâce à la classe DateTime.



- Définir les propriétés comme ci-dessous:

Propriété	Valeur
Name	labelHorloge
AutoSize	False
BorderStyle	FixedSingle
Location	96;120
Size	100;23
Text	
TextAlign	MiddleCenter

AutoSize permet à la taille du label de s'ajuster automatiquement à la longueur du texte, ici nous n'utilisons pas cette fonctionnalité.

IV) Création du gestionnaire d'événements.

Double cliquer sur le contrôle timerHorloge afin de visualiser le gestionnaire d'événement tick et compléter le comme ci-dessous:

```
private void timerHorloge_Tick(object sender, EventArgs e)
{
    labelHorloge.Text = DateTime.Now.ToLongTimeString();
}
```

V) Compiler et tester le projet.

VI) La classe DateTime

Pour des précisions sur la classe DateTime et la manipulation de dates et d'heures, consulter les pages 184 à 193 inclus.

VI) Améliorations

- Compléter l'application suivante afin d'afficher aussi la date.
- Compléter l'application afin d'ajouter un chronomètre, suivre les indications suivantes.

1° version

Un bouton start démarre le chronomètre.

Un bouton arrête le compte.

La valeur s'affiche dans un champ label uniquement lors de l'appui sur le bouton Stop.

Utiliser la classe Stopwatch pour récupérer le nombre de millisecondes écoulées.

Traiter ce nombre (calculs simples) afin d'extraire les secondes



2° version

Même chose que la 1° version, mais on voit la valeur du chronomètre défiler.

Pour obtenir ces informations (noms de mois) pour la « culture » courante :

```
string[] ts = CultureInfo.CurrentCulture.DateTimeFormat.MonthNames;
```

Le tableau de mois correspondant à MonthNames comprend treize entrées, la dernière étant vide.
Le tableau correspondant à DayNames comprend sept cellules, avec Dimanche dans la première.

2.4.4 Mesure d'intervalles de temps

Pour mesurer un intervalle de temps, il est possible d'écrire :

```
DateTime t1 = DateTime.Now;           // temps de départ
.....                               // exécuter des instructions
DateTime t2 = DateTime.Now;           // temps d'arrivée
TimeSpan ts = t2 - t1;
```

ts.TotalMilliseconds donne alors la durée, exprimée en millisecondes, entre les deux événements.

Depuis la version 2, .NET a introduit la classe Stopwatch (de l'espace de noms System.Diagnostics) afin de mesurer des temps avec grande précision.

Classe Stopwatch

```
StopWatch ← ValueType ← Object
```

```
using System.Diagnostics;
```

Propriétés de la classe Stopwatch

Elapsed	TimeSpan	Durée de type TimeSpan entre Start et Stop.
ElapsedMilliseconds	long	Durée, en millisecondes.

Méthodes de la classe Stopwatch

Start();	Démarre le chronomètre.
Stop();	Arrête le chronomètre.
bool IsRunning();	Indique si le chronomètre est actif.

On écrira par exemple :

```
using System.Diagnostics;
StopWatch sw = new Stopwatch();
sw.Start();                               // démarrer le chronomètre
.....
sw.Stop();                               // l'arrêter
```

La propriété IsHighResolution, de type bool, indique si la résolution haute a pu être utilisée pour mesurer des temps. sw.ElapsedMilliseconds donne alors le nombre de millisecondes qui se sont écoulées entre Start et Stop.