



Concepteur Développeur en Informatique

Développer des composants d'interface

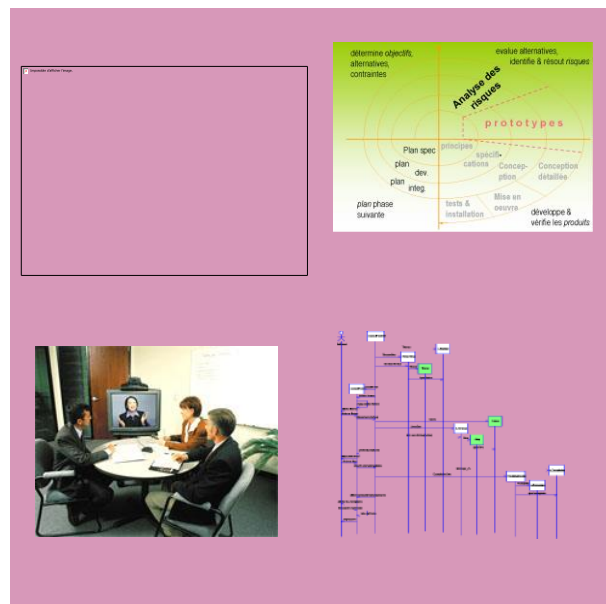
Elaborer des formulaires Windows Menus

Accueil

Apprentissage

PAE

Evaluation



Localisation : U03-E03-S02

SOMMAIRE

1	Introduction	3
2	Le menu de l'application (MenuStrip)	3
2.1	Gestion des évènements	5
3	Les menus contextuels (ContextMenuStrip)	6
4	La barre d'outils (ToolStrip)	7
5	La barre de statut (StatusStrip)	8
6	Le ToolStripContainer	9
	

1 Introduction

Les contrôles **ToolStrip**, **MenuStrip**, **StatusStrip** et **ContextMenuStrip** remplacent respectivement les contrôles **ToolBar**, **MainMenu**, **StatusBar** et **ContextMenu** qui sont encore présents pour préserver la compatibilité ascendante.

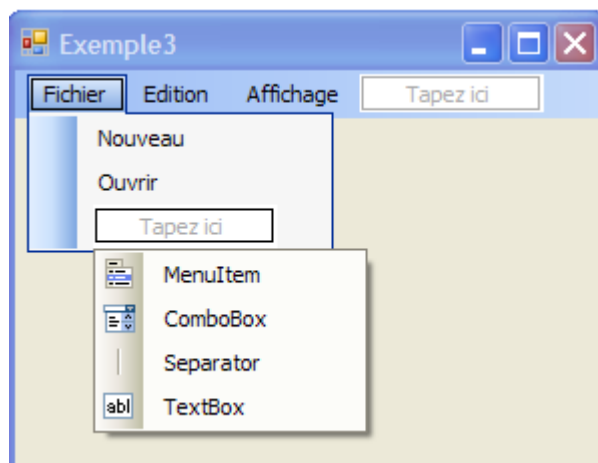
Ils sont regroupés dans l'onglet Menu et barres d'outils de la boîte à outils.



En plus d'un aspect visuel plus travaillé, de nouvelles fonctionnalités ont été ajoutées, telles que **l'organisation et l'alignement de texte et d'images**.

Il suffit de double-cliquer sur un de ces composants pour l'ajouter au formulaire.

2 Le menu de l'application (MenuStrip)

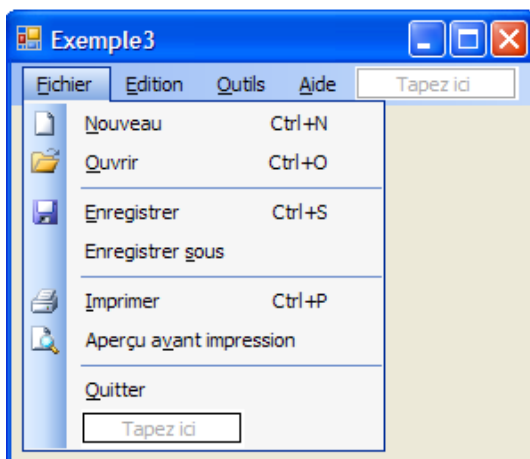


Un élément (instance de la classe **ToolStripMenuItem**) est créé en tapant son libellé directement dans l'éditeur de Menus (dans la zone *Tapez ici*).

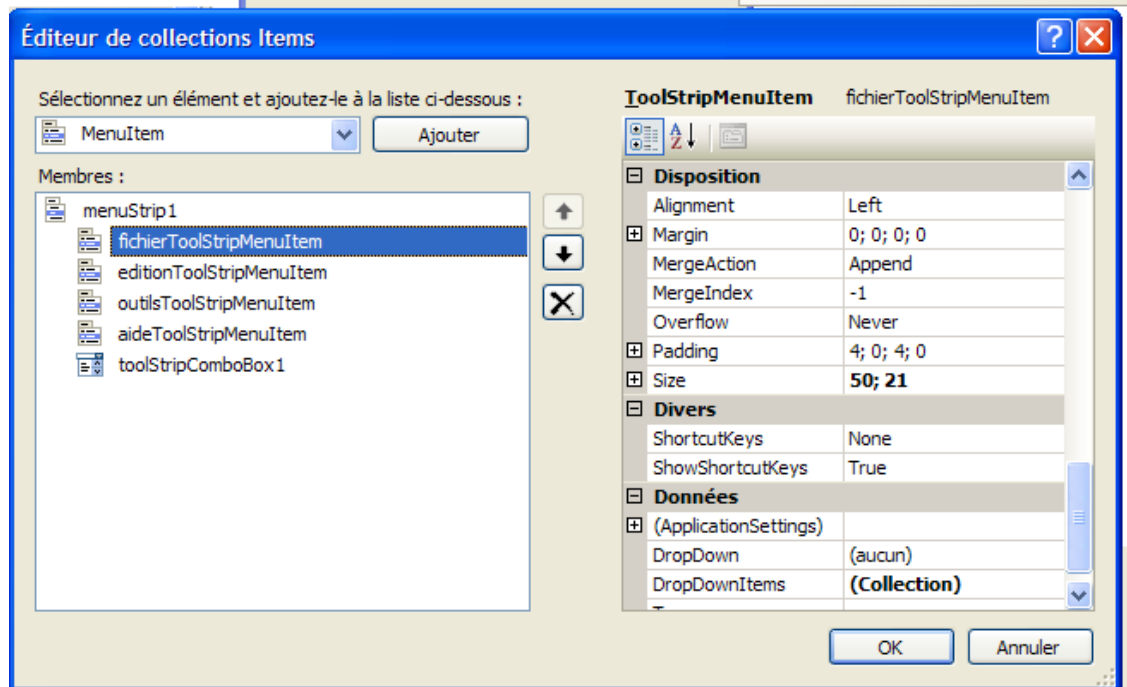
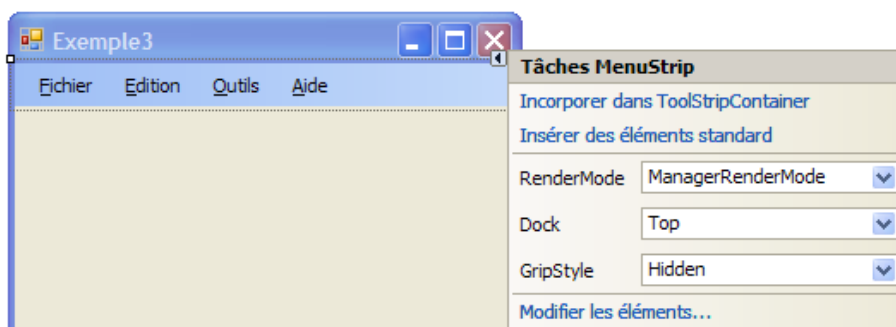
La nature de l'élément (**MenuItem**, **ComboBox**, **Separator**, **TextBox**) peut être choisie en cliquant sur l'élément

Le menu principal stocke sa collection d'élément dans une collection appelée *Items*, les sous-menus utilisent une collection **DropDownItems**.

L'option *Insérer des éléments standards* de la liste des tâches du menu principal permet d'insérer un menu standard.



L'option *Modifier les éléments* de la liste des tâches du menu principal permet de modifier les propriétés de chaque élément (faisable également dans la fenêtre de propriétés directement)



Chaque élément de menu possède les propriétés remarquables suivantes :

- **Text** Libellé de l'élément (& devant la lettre pour actionner le raccourci)
- **ShortcutKey** Touche de raccourci associée à l'élément
- **ShowShortcutKeys** Indique si la touche de raccourci est affichée sur l'élément
- **TextDirection** Permet de fixer la direction du libellé du menu
- **TextImageRelation** Spécifie la position du libellé par rapport à l'image
- **Image** Définit l'image qui sera affichée sur l'élément
- **Checked** Indique que l'élément est coché
- **Enabled** Indique que le menu est cliquable
- **Index** Position de l'élément dans le menu
- **Visible** Pour cacher l'élément

2.1 Gestion des événements

Chaque élément de menu peut générer un événement *Click*.

Le gestionnaire de l'évènement Click sera obtenu en double-cliquant soit sur le libellé du menu, soit sur l'évènement Click correspondant au menu à traiter dans la fenêtre des propriétés.

```
private void nouveauToolStripMenuItem_Click(object sender, EventArgs e)
{
}
```

On peut également ajouter dynamiquement des éléments de menus au menu existant :

1. Ajouter dans le programme une fonction de traitement, ou utilisez une fonction déjà créée.

```
private void mnu_Click(object sender, System.EventArgs e)
{
}
```

2. Pour ajouter l'élément Fermer en fin de menu Fichier, et faire traiter cet élément par la fonction btnQuitter_Click, il suffira de coder :

```
ToolStripMenuItem tSMI = new ToolStripMenuItem();
tSMI.Text=" Fermer ";
tSMI.Click += new System.EventHandler(this.btnQuitter_Click);
this.mnuFichier.DropDownItems.Add(tSMI);
```

Si la fonction de traitement traite plusieurs points de menus, on pourra détecter l'origine du click en écrivant :

```
ToolStripMenuItem tSMI = (ToolStripMenuItem)sender ;
if (tSMI.Text == "Fermer ") ...
```

Il n'existe aucun événement pour signaler qu'un menu a été fermé.

Vous devrez gérer les événements de la classe Form suivants pour envisager de contrôler les accès aux options de menus :

- **MenuStart** le menu reçoit le focus
- **MenuComplete** le menu perd le focus

3 Les menus contextuels (ContextMenuStrip)

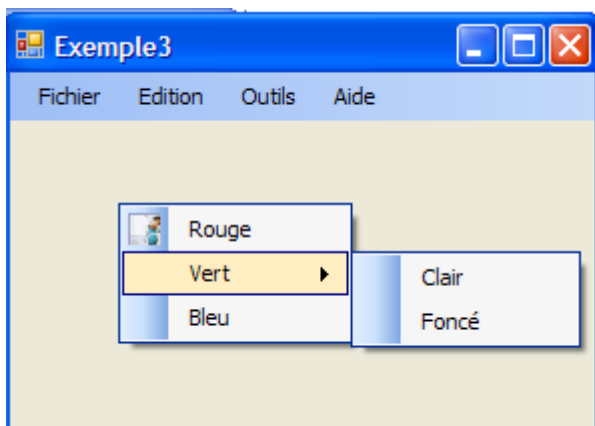
Importé directement de la boîte à outils, c'est un objet de la classe **ContextMenuStrip**, dérivée de **MenuStrip**.

Il a donc toutes les caractéristiques d'un menu.

Il est créé et traité comme un menu, et permet d'associer un menu contextuel à n'importe quel contrôle.

La classe **ContextMenuStrip** représente des menus contextuels qui sont affichés lorsque l'utilisateur clique avec le bouton droit de la souris sur un contrôle ou une zone du formulaire.

Les menus contextuels sont généralement utilisés pour combiner différents éléments de menu d'un **MenuStrip** d'un formulaire qui sont utiles à l'utilisateur compte tenu du contexte de l'application.



L'éditeur de collection Items dans la fenêtre de propriétés permet d'ajouter facilement tout élément à la barre d'outils et d'en modifier ses propriétés (accessible également dans la liste des tâches par l'option *Modifier les éléments*)

Les propriétés **ShowCheckMargin** et **ShowImageMargin** permettent de laisser apparaître de l'espace pour visualiser une image ou une coche lorsque l'option a été sélectionnée.

Pour afficher le menu contextuel à l'emplacement de la souris, lorsque l'utilisateur clique sur la feuille, on pourra coder :

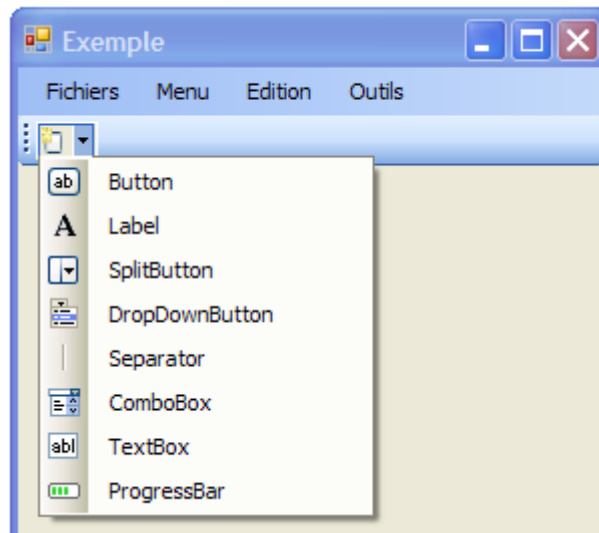
```
private void Exemple3_MouseDown(object sender, MouseEventArgs e)
{
    contextMenuCouleur.Show(this, new Point(e.X, e.Y));
}
```

4 La barre d'outils (ToolStrip)

Le composant **ToolStrip** remplace et ajoute des fonctionnalités au contrôle **ToolBar** des versions antérieures,

ToolBar est conservé pour la compatibilité descendante et l'utilisation éventuelle.

Le composant **ToolStrip** permet de créer rapidement une barre d'outils en incorporant des objets divers :



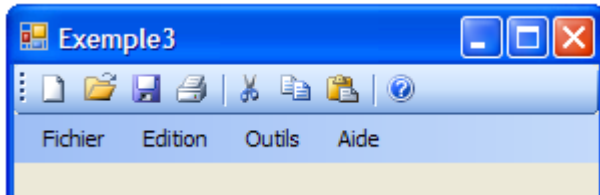
- Un **ToolStripButton** permet d'incorporer du texte et une image.

Utilisez les propriétés `ToolStripItem.ImageAlign` et `ToolStripItem.TextAlign` pour obtenir ou définir la position des images et du texte **ToolStripButton**

- Un **ToolStripLabel** figure un **ToolStripItem** non sélectionnable qui restitue texte et images.
- Un **ToolStripSplitButton** est une combinaison d'un bouton standard à gauche et d'une liste déroulante à droite ou inversement (si la valeur de **RightToLeft** est **Yes**).
- Un **ToolStripDropDownButton** représentant un contrôle qui, lorsque l'utilisateur clique dessus, affiche un **ToolStripDropDown** associé à partir duquel l'utilisateur peut sélectionner un seul élément.
- Un **ToolStripSeparator** permet l'insertion d'une ligne utilisée pour grouper des éléments.
- Un **ToolStripComboBox** affiche un champ d'édition associé à un objet **ListBox**. Il permet à l'utilisateur de sélectionner un élément dans la liste ou d'en insérer un nouveau.
- Un **ToolStripTextBox** représente une zone de texte dans un **ToolStrip** qui permet à l'utilisateur d'entrer du texte
- Un **ToolStripProgressBar** représente un contrôle de barre de progression Windows contenu dans **StatusStrip**.

L'éditeur de collection Items dans la fenêtre de propriétés permet d'ajouter facilement tout élément à la barre d'outils et d'en modifier ses propriétés (accessible également dans la liste des tâches par l'option *Modifier les éléments*)

Comme pour le **MenuStrip**, l'option *Insérer des éléments standards* de la liste des tâches de la barre d'outils permet de créer automatiquement les boutons d'une barre d'outils standard.



La barre d'outils standard générée automatiquement

Chaque élément de la barre peut générer un évènement *Click*.

Le gestionnaire de l'évènement Click sera obtenu en double-cliquant soit sur le bouton de la barre, soit sur l'évènement Click correspondant au bouton à traiter dans la fenêtre des propriétés.

En cliquant sur l'icône Enregistrer, on obtient:

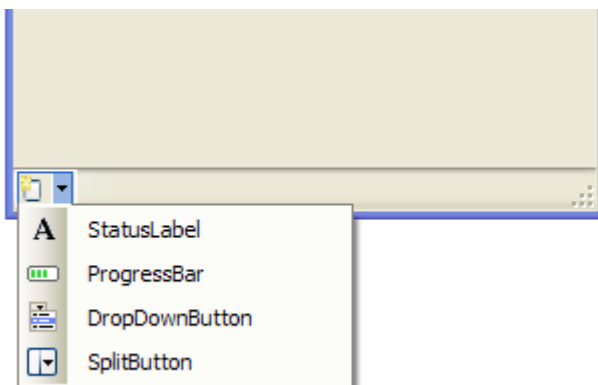
```
private void enregistrerToolStripButton_Click(object sender,
EventArgs e)
{
}
```

5 La barre de statut (StatusStrip)

Le composant **StatusStrip** remplace et élargit le contrôle **StatusBar** des versions antérieures, **StatusBar** est conservé pour la compatibilité descendante.

Le composant **StatusStrip** représente une barre de statut qui apparaît généralement en bas de la fenêtre. En règle générale, un contrôle **StatusStrip** est constitué d'objets **ToolStripStatusLabel**, dont chacun affiche du texte et/ou une icône.

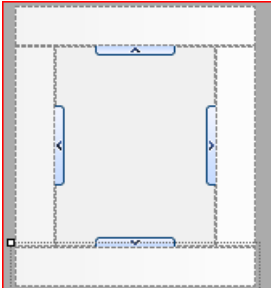
Le **StatusStrip** peut également contenir des contrôles **ToolStripDropDownButton**, **ToolStripSplitButton** et **ToolStripProgressBar**.



L'éditeur de collection Items dans la fenêtre de propriétés permet d'ajouter facilement tout élément à la barre de statut et d'en modifier ses propriétés (accessible également dans la liste des tâches par l'option *Modifier les éléments*)

6 Le ToolStripContainer

Le contrôle **ToolStripContainer** (System.Windows.Forms.ToolStripContainer) sert de conteneur pour les barres d'outils. Vous pouvez organiser et modifier les positions des barres d'outils à l'intérieur de ce conteneur. **ToolStripContainer** est composé de 4 **ToolStripPanels** visible de chaque côté (Haut, Bas, Gauche, Droite) où vous pouvez placer vos **ToolStrips**.



Plusieurs contrôles **ToolStrip** s'empilent verticalement si vous les placez dans le **ToolStripContainer** gauche ou droit.

Ils s'empilent horizontalement si vous les placez dans le **ToolStripContainer** supérieur ou inférieur.

