



Module 1 Développer l'interface d'une application informatique

UI Windows

Séance

S05

Activité

A-002

Cette activité vous permettra de travailler sur deux thèmes :

- La construction d'un dialogue modal
- La validation des champs de formulaire.

Sommaire :

1	Mise en œuvre d'un dialogue modal	2
2	Dialogue de connexion	3
2.1	Modification des propriétés du formulaire	3
2.2	Validation des données du formulaire.....	3

1 Mise en œuvre d'un dialogue modal

Un dialogue modal est mis en œuvre via une fenêtre particulière désignée sous le terme de boîte de dialogue. Nous allons ici mettre en place une boîte de dialogue personnalisée.

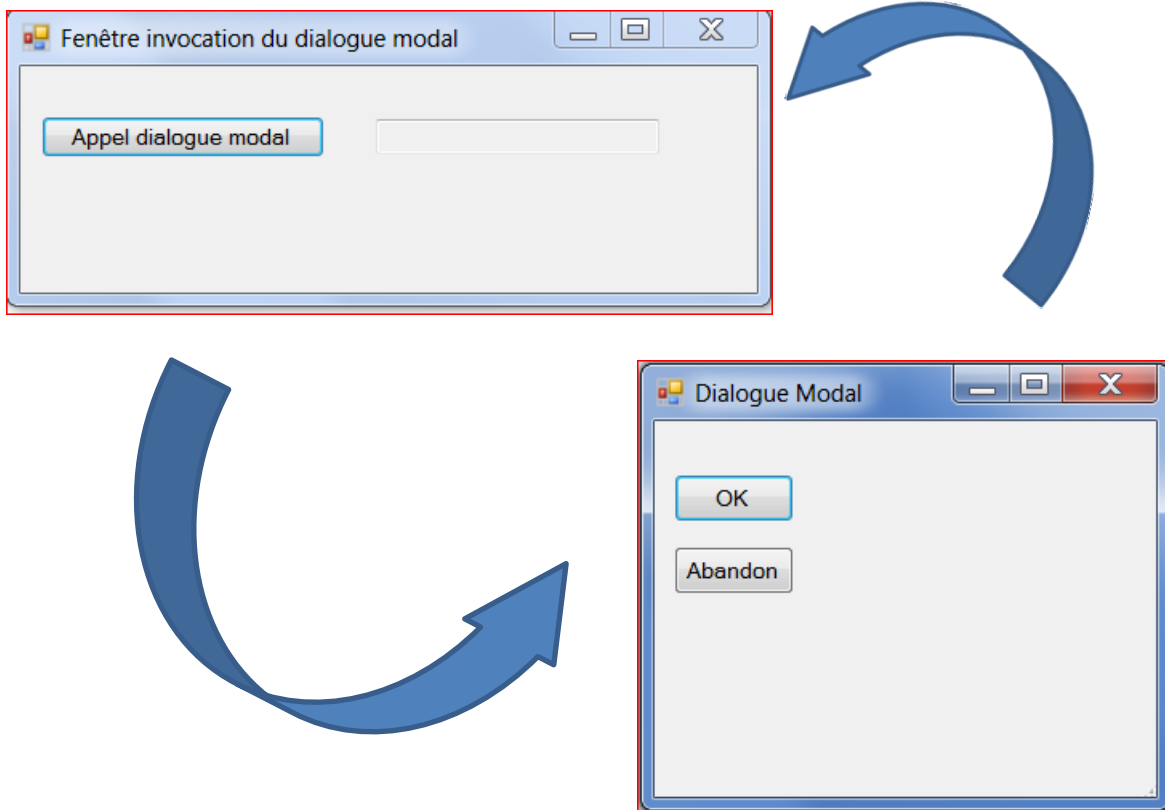
Référez-vous au point 6.3 du document présentant la programmation sous Windows.

Une fenêtre « modale » comporte des boutons permettant de terminer le dialogue et de récupérer, au point de retour de la méthode **ShowDialog()** de l'instance de la fenêtre, le résultat du dialogue.

Ce résultat est une valeur parmi la liste énumérée (Enum) **DialogResult**.

Dans ce premier exemple, nous disposons d'une fenêtre Windows qui permet d'invoquer la fenêtre de dialogue modal et de traiter le résultat du dialogue.

```
FrmdialogueModal dialogueModal = new FrmdialogueModal();  
DialogResult resultat = dialogueModal.ShowDialog();
```



Définition de l'énumération DialogResult et la liste des choix associée (extrait) :

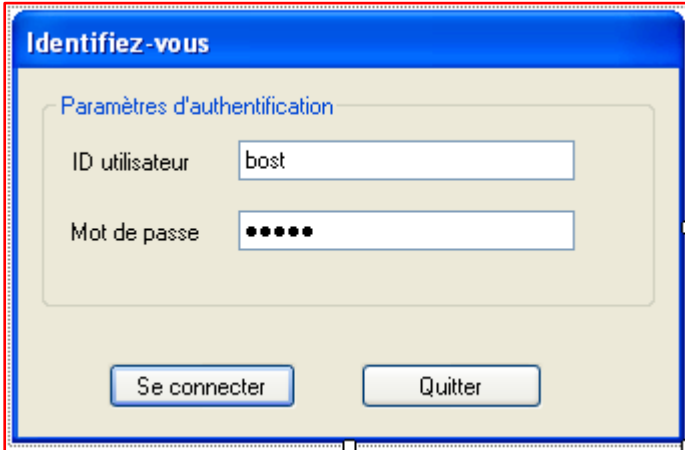
```
...public enum DialogResult  
{  
    ...None = 0,  
    ...OK = 1,  
    ...Cancel = 2,  
    ...Abort = 3,  
    ...Retry = 4,  
    ...Ignore = 5,  
    ...Yes = 6,  
    ...No = 7,  
}
```

```
switch (resultat)  
{  
    case DialogResult.Abort:  
        txtResultat.Text = "Abandon";  
        break;  
    case DialogResult.Cancel:  
        txtResultat.Text = "Annulation";  
        break;  
    case DialogResult.Ignore:  
        break;  
}
```

2 Dialogue de connexion

Vous devez mettre en œuvre un dialogue qui autorise ou non l'accès d'un utilisateur à l'application qui le met en œuvre.

La fenêtre du dialogue devra avoir une apparence semblable à celle-ci :



Ce dialogue modal comporte deux conditions de sortie qui devront être testées au point de retour de l'appel de la méthode ShowDialog().

DialogResultOK est la valeur retournée si l'utilisateur a cliqué sur le bouton du bouton « Se connecter » ou **ENTREE**

DialogResultCancel est la valeur retournée si l'utilisateur a choisi « Quitter » ou la touche **ECHAP**

2.1 Modification des propriétés du formulaire

Après avoir défini en Design Time (Conception) les boutons Se Connecter et Quitter, assigner les aux propriétés **AcceptButton** et **CancelButton** du formulaire afin de prendre en charge les touches **Enter** et **Escap** sans mettre en œuvre de traitements complexes des événements de clavier.

Une boîte de dialogue ne dispose pas du menu système et ne peut être redimensionnée.

Positionnez la fenêtre au centre de la fenêtre.

2.2 Validation des données du formulaire

Vous pouvez sortir de la boîte de dialogue et revenir au point d'appel selon deux procédés :

1. Vous avez cliqué sur le bouton **Quitter** ou appuyé sur la touche **ECHAP**.
Dans ce cas de figure, vous devez sortir du dialogue avec une réponse de type **DialogResult.Cancel**.
2. Vous avez cliqué sur le bouton **Se connecter** ou appuyé sur la touche **ENTREE**.
Si aucune erreur ne subsiste sur le formulaire (les contrôles sont corrects), alors vous devez sortir du dialogue avec une réponse de type **DialogResult.OK**.
Si des erreurs subsistent vous devez poursuivre le dialogue et annuler la sortie de ce dernier.

Il vous faudra alors assigner à DialogResult la valeur **DialogResult.None**.

Si vous avez des contrôles de validité de champs de formulaire, ces contrôles ne doivent pas être effectués lorsque l'utilisateur abandonne sa saisie. La demande de validation se produit lorsque la propriété **CausesValidation** de la cible vaut **true**. La propriété **CausesValidation** du bouton **Quitter** doit donc mis à false.

Le contrôle des champs se fait sur l'événement **Validating** de chaque boîte de texte qui se produit lors de la sortie du champ de formulaire (avant perte du focus). Tant que la valeur saisie n'est pas correcte, on ne peut passer au contrôle suivant.

Pour signaler les erreurs à l'utilisateur, utilisez le contrôle **ErrorProvider**.

Règles de vérification des données :

1. L'identifiant est obligatoire
2. L'identifiant peut contenir des caractères alpha et numériques mais doit commencer par une lettre.
3. La longueur de l'identifiant ne peut être inférieure à 5
4. Le mot de passe est obligatoire
5. Longueur minimale du mot de passe 5
6. Mot de passe = Identifiant (pour le moment !)