

Arguments d'appel d'une application

L'usage des arguments d'appel d'une application permet à l'utilisateur de lancer le programme en lui passant des paramètres. Par exemple, la commande **Format** du Dos nécessite le passage d'un paramètre désignant l'unité à formater. De même, la commande **Copy** a besoin du nom du fichier à copier et du nom du fichier de destination.

```
Format D:  
Copy MonFichier.Txt MonFichier.Sav
```

Ces commandes travaillent ainsi parce que leurs arguments d'appel ont été déclarés lors de l'écriture de la procédure **Main**, où ils sont aussi appelés paramètres formels, et que les paramètres passés en ligne de commande, aussi appelés paramètres effectifs, leur sont substitués lors de l'exécution du code. Ces paramètres se déclarent comme étant un tableau de **String**. A l'exécution, chaque paramètre effectif occupe une place dans ce tableau.

Application Console

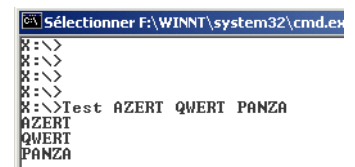
Le mécanisme des paramètres en ligne de commande impose qu'une procédure **Main** soit écrite et désignée comme objet de démarrage. C'est la signature choisie pour cette **Sub Main** qui permet ou interdit la prise en compte de paramètres effectifs.

```
Sub Main()  
Sub Main(ByVal Arguments() As String)
```

' Sans paramètre
' Avec paramètre(s)

Le code suivant est compilé pour fournir le programme **Test.exe**.

```
Module MonModule  
Sub Main(ByVal Arguments() As String)  
For i As Integer = 0 To Arguments.Length - 1  
Console.WriteLine(Arguments(i))  
Next  
Console.ReadLine()  
End Sub  
End Module
```



Application Windows

Certaines applications Windows peuvent être lancées à partir d'une session Dos, d'autres seulement à partir de l'interface graphique, par un raccourci ou par un double-clic de leur nom dans l'*Explorateur* de Windows. La prise en compte des paramètres se programme différemment selon ces cas.

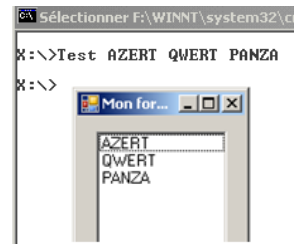
A partir d'une session Dos

L'application est initialement construite comme étant une application Console. Elle est composée du module **MonModule** et du formulaire **FAffichage** sur lequel une **ListBox** nommée **MaListBox** a été dessinée. La **Sub Main** a été désignée comme objet de démarrage dans les propriétés du projet. L'application compilée donne le programme **Test.exe**.

```

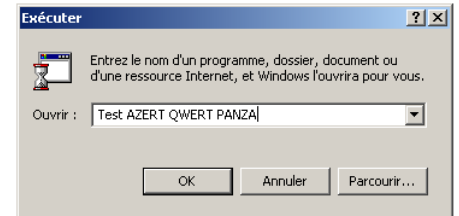
Sub Main(ByVal Arguments() As String)
    Dim F As New FAffichage
    For i As Integer = 0 To Arguments.Length - 1
        F.MaListBox.Items.Add(Arguments(i))
    Next
    F.ShowDialog()
End Sub

```



A partir de Windows

L'application est construite comme étant une application Windows. Elle est uniquement composée du formulaire **FBase** sur lequel une **ListBox** nommée **MaListBox** a été dessinée. L'application compilée donne le programme **Test.exe**. Il peut être activé comme n'importe quelle autre application Windows. Toutefois, pour lui passer des paramètres d'exécution, le programme est activé à partir de la boîte de dialogue *Exécuter* de Windows.



C'est la librairie **Microsoft.VisualBasic** qui fournit l'outil nécessaire, la fonction **Command**, qui livre une chaîne de caractères composées des paramètres d'exécution. La méthode **Split** de la classe **String** permet la transformation de cette chaîne en un tableau où chaque paramètre est disponible distinctement. La prise en compte des paramètres d'exécution se programme ordinairement comme ci-dessous, dans la procédure de réponse à l'événement **Load** du formulaire de démarrage.

```

Private Sub FBase_Load(ByVal sender As Object,
                        ByVal e As System.EventArgs) Handles Me.Load
    Dim Arguments() As String =
        Microsoft.VisualBasic.Command.Split(" ")
    For i As Integer = 0 To Arguments.Length - 1
        MaListBox.Items.Add(Arguments(i))
    Next
End Sub

```

