If...Then...Else Statements

Visual Studio .NET 2003

Conditionally executes a group of statements, depending on the value of an expression.

```
If condition [ Then ]
   [ statements  ]
[ ElseIf elseifcondition [ Then ]
   [ elseifstatements ] ]
[ Else
   [ elsestatements ] ]
End If
```

-or-

```
If condition Then [ statements ] [ Else elsestatements ]
```

Parts

condition
Required. Expression. The expression you supply for *condition* must evaluate to **True** or **False**, or to a data type that is implicitly convertible to **Boolean**.

statements
Optional in multiple-line form; required in single-line form that has no **Else** clause. One or more statements following **If...Then** that are executed if *condition* is **True**.

elseifcondition
Required if **ElseIf** is present. Same as *condition*.

elseifstatements
Optional. One or more statements following **ElseIf...Then** that are executed if the associated *elseifcondition* is **True**.

elsestatements
Optional in multiple-line form; required in single-line form that has an **Else** clause. One or more statements that are executed if no previous *condition* or *elseifcondition* expression is **True**.

End If
Terminates **If...Then** block.

Remarks

You can use the single-line form for short, simple tests. However, the multiple-line form provides more structure and flexibility than the single-line form and is usually easier to read, maintain, and debug.

With the single-line form, it is possible to have multiple statements executed as the result of an **If...Then** decision. All statements must be on the same line and be separated by colons, as in the following example:

```
If A > 10 Then A = A + 1 : B = B + A : C = C + B
```

In the multiple-line form, the **If** statement must be the only statement on the first line. The **Else**, **ElseIf**, and **End If** statements can be preceded only by a line label. The multiple-line **If...Then...Else** must end with an **End If** statement.

To determine whether or not an **If** statement introduces a multiple-line form, examine what follows the **Then** keyword. If anything other than a comment appears after **Then** in the same statement, the statement is treated as a single-line **If** statement. If **Then** is absent, it must be the beginning of a multiple-line **If...Then...Else**.

The **ElseIf** and **Else** clauses are both optional. You can have as many **ElseIf** clauses as you want in a multiple-line **If...Then...Else**, but none can appear after an **Else** clause. Multiple-line forms can be nested within one another.

When a multiple-line **If...Then...Else** is encountered, *condition* is tested. If *condition* is **True**, the statements following **Then** are executed. If *condition* is **False**, each **ElseIf** statement is evaluated in order. When a **True** *elseifcondition* is found, the statements immediately following the associated **Then** are executed. If no *elseifcondition* evaluates to **True**, or if there are no **ElseIf** statements, the statements following **Else** are executed. After executing the statements following **Then**, **ElseIf**, or **Else**, execution continues with the statement following **End If**.

> **Tip** **Select Case** might be more useful when evaluating a single expression that has several possible values.

Example

This example shows both the multiple- and single-line forms of the **If...Then...Else** statement.

```
Dim Number, Digits As Integer
Dim MyString As String
Number = 53   ' Initialize variable.
If Number < 10 Then
   Digits = 1
ElseIf Number < 100 Then
' Condition evaluates to True so the next statement is executed.
   Digits = 2
Else
   Digits = 3
End If

' Assign a value using the single-line form of syntax.
If Digits = 1 Then MyString = "One" Else MyString = "More than one"
```

Use the **TypeOf** keyword to determine whether the **Control** object passed into a procedure is a text box.

```
Sub ControlProcessor(ByVal MyControl As Control)
```

```
    If TypeOf MyControl Is ComboBox Then
       Debug.WriteLine ("You passed in a " & TypeName(MyControl))
    ElseIf TypeOf MyControl Is CheckBox Then
       Debug.WriteLine ("You passed in a " & TypeName(MyControl))
    ElseIf TypeOf MyControl Is TextBox Then
       Debug.WriteLine ("You passed in a " & TypeName(MyControl))
    End If
End Sub
```

See Also

#If...Then...#Else Directives | Choose Function | Select...Case Statements | Switch Function | If...Then...Else Statements (Conceptual)