

10/10/2010



Interfaces

Distributed Enterprise Applications



Stephen Hole

Interfaces

An interface is a set of properties and methods that a class exposes. It is a contract that guarantees to a client how a class will behave. An interface defines only the signature of such properties and methods (member name, number and type of each parameter, and type of return value), while a class can implement that interface by providing actual code for those properties and methods as necessary. When a class implements an interface, it must implement all parts of that interface; in effect the class says *“I agree to fulfil the contract defined by this interface.”* The convention for naming interfaces is to start the name with a capital I. This convention is not enforced by Visual Studio but is most useful for identifying interface objects.

Defining the Interface

- Create a new Windows Form application and name it *InterfaceExample*.
- Add a class module name *Document* to the project.
- Add the following Import statements:

```
Imports System
Imports System.Collections.Generic
Imports System.Text
```

- Now create the following interface directly below the import statements:

```
Interface ISStorable
    Sub ReadMethod()
    Sub WriteMethod(ByVal o As Object)
    Property Status()
End Interface
```

- Add the phrase Implements ISStorable to the class definition as show below:

```
Public Class Document : Implements ISStorable
```

When you press the enter key the code stubs for the functions and property are automatically added to the class module.

```
Public Class Document : Implements ISStorable

    Public Sub ReadMethod() Implements ISStorable.ReadMethod
    End Function

    Public Property Status As Object Implements ISStorable.Status
        Get

            End Get
            Set(ByVal value As Object)

            End Set
        End Property

    Public Sub WriteMethod(ByVal o As Object) Implements ISStorable.WriteMethod
    End Function
End Class
```

Implementing the Interface

```
'Create the class that implements the ISStorable
Public Class Document : Implements ISStorable
    'Store the value for the property
    Private myStatus As Integer = 0

    Public Sub New(ByVal s As String)
        MessageBox.Show("Creating Document with: " & s)
    End Sub

    'Implement the ReadMethod
    Public Sub ReadMethod() Implements ISStorable.ReadMethod
        MessageBox.Show("Implementing the ReadMethod for the ISStoreable interface")
    End Sub

    Public Property Status As Object Implements ISStorable.Status
        Get
            Return myStatus
        End Get
        Set(ByVal value As Object)
            myStatus = value
        End Set
    End Property

    Public Sub WriteMethod(ByVal o As Object) Implements ISStorable.WriteMethod
        MessageBox.Show("Implementing the WriteMethod for the ISStoreable interface")
    End Sub
End Class
```

Testing the Interface

- Add a button named InterfaceTestButton to the Windows Form and place the following code in the Click event.

```
Private Sub InterfaceTestButton_Click(...) Handles InterfaceTestButton.Click
    Dim doc As New Document("Test Document")
    doc.Status = -1
    doc.ReadMethod()
    MessageBox.Show("Document Status: " & doc.Status)
End Sub
```

- Run the application a test the interface making a note of the output.

Implementing More Than One Interface

Classes can implement more than one interface. In addition to the ISStorable interface the ICompressible interface can implemented by the Document class as shown below:

```
Imports System
Imports System.Collections.Generic
Imports System.Text

Interface ISStorable
    Sub ReadMethod()
    Sub WriteMethod(ByVal o As Object)
    Property Status()
End Interface

Interface ICompressible
    Sub Compress()
    Sub Decompress()
End Interface

'Create the class that implements the ISStorable and ICompressible Interfaces
Public Class Document : Implements ISStorable, ICompressible
    'Store the value for the property
    Private myStatus As Integer = 0

    Public Sub New(ByVal s As String)
        MessageBox.Show("Creating Document with: " & s)
    End Sub

    'Implement the ReadMethod
    Public Sub ReadMethod() Implements ISStorable.ReadMethod
        MessageBox.Show("Implementing the ReadMethod for the ISStoreable interface")
    End Sub

    Public Property Status As Object Implements ISStorable.Status
        Get
            Return myStatus
        End Get
        Set(ByVal value As Object)
            myStatus = value
        End Set
    End Property

    Public Sub WriteMethod(ByVal o As Object) Implements ISStorable.WriteMethod
        MessageBox.Show("Implementing the WriteMethod for the ISStoreable interface")
    End Sub

    'Implement the ICompressible Interface
    Public Sub Compress() Implements ICompressible.Compress
        MessageBox.Show("Implementing the Compress Method for the ICompressible interface")
    End Sub

    Public Sub Decompress() Implements ICompressible.Decompress
        MessageBox.Show("Implementing the Decompress Method for the ICompressible interface")
    End Sub
End Class
```

Extending Interfaces

It is possible to extend an existing interface to add new methods or members, or to modify how existing members work. The `ICompressible` interface can be extended with a new interface named `ILoggedCompressible`:

```
Imports System
Imports System.Collections.Generic
Imports System.Text

Interface ISStorable
    Sub ReadMethod()
    Sub WriteMethod(ByVal o As Object)
    Property Status()
End Interface

Interface ICompressible
    Sub Compress()
    Sub Decompress()
End Interface

Interface ILoggedCompressible
    Inherits ICompressible
    Sub LogSavedBytes()
End Interface

'Create the class that implements the ISStorable and ICompressible Interfaces
Public Class Document : Implements ISStorable, ICompressible, ILoggedCompressible
    'Store the value for the property
    Private myStatus As Integer = 0

    Public Sub New(ByVal s As String)
        MessageBox.Show("Creating Document with: " & s)
    End Sub

    'Implement the ReadMethod
    Public Sub ReadMethod() Implements ISStorable.ReadMethod
        MessageBox.Show("Implementing the ReadMethod for the ISStoreable interface")
    End Sub

    Public Property Status As Object Implements ISStorable.Status
        Get
            Return myStatus
        End Get
        Set(ByVal value As Object)
            myStatus = value
        End Set
    End Property

    Public Sub WriteMethod(ByVal o As Object) Implements ISStorable.WriteMethod
        MessageBox.Show("Implementing the WriteMethod for the ISStoreable interface")
    End Sub

    'Implement the ICompressible Interface
    Public Sub Compress() Implements ICompressible.Compress
        MessageBox.Show("Implementing the Compress Method for the ICompressible interface")
    End Sub

    Public Sub Decompress() Implements ICompressible.Decompress
        MessageBox.Show("Implementing the Decompress Method for the ICompressible interface")
    End Sub

    'Implementing the ILoggedCompressible Interface
    Public Sub LoggedSavedBytes() Implements ILoggedCompressible.LogSavedBytes
        MessageBox.Show("Implementing the LoggedSavedBytes Method for the ILoggedCompressible interface")
    End Sub
End Class
```

The ICompressible methods (Compress and Decompress) can either be implemented from ICompressible itself or from the extended interface ILoggedCompressible; see the highlighted code.

```
'Implement the ICompressible Interface
Public Sub Compress() Implements ICompressible.Compress
    MessageBox.Show("Implementing the Compress Method for the ICompressible interface")
End Sub

Public Sub Decompress() Implements ILoggedCompressible.Decompress
    MessageBox.Show("Implementing the Decompress Method for the ILoggedCompressible interface")
End Sub

'Implementing the ILoggedCompressible Interface
Public Sub LoggedSavedBytes() Implements ILoggedCompressible.LogSavedBytes
    MessageBox.Show("Implementing the LoggedSavedBytes Method for the ILoggedCompressible interface")
End Sub
```