

Architecting Dilemma

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vb/2007-04/msg00652.html>

- *From:* "John Wright" <riley_wrightx@xxxxxxxxxxxx>
 - *Date:* Tue, 10 Apr 2007 15:36:09 -0600
-

Here is the situation. We are rearchitecting an application from VB 6.0 to VB.NET 2005. Of course in VB6 there was no OOP so there are lots of CLS files that do the work for each object type. I am proposing creating a base class object that contains all the common properties and methods for an object, then using inheritance, creating specific types of objects from the base class. This is not a problem. This is a manufacturing application and the manufacturing process is FOO ----> Widget ----> Piece. All Widgets are created from FOO's and all Pieces are created from Widgets. I have proposed creating a Widget base class and inheriting the Piece classes from Widget. Widget will contain the FOO ID and FOO Part Number as read only properties since they are really the only thing needed for the Piece class. Not a problem. So to create a FOO I do the following (pseudocode)

```
Dim x as new FOO
x.ID = 5
x.PartNumber = "12345B"
x.Weight = 2500
x.Traveler = "FOO 124(4)"
x.Standard = "FOO 344-(J)"
x.ChemistryID = 44
x.update.
```

This creates a new FOO item. Next I need to create a Piece from Widget so pseudocoding again, I create a Piece Class (inheriting from Widget):

```
Dim Y as new Piece(54)
y.traveler = "Piece12345A" 'Set in Widget since all pieces have a traveler
y.Standard = "Standard1234" 'Set in Widget since all pieces have a standard
y.weight = 32 'Set in the Piece class since not all pieces have a weight
y.Step = 44 'Set in Widget since all pieces have a traveler step
y.PartNumber = "PieceX" 'Set in Widget since all pieces have a PartNumber
y.Batch = "44x" 'Set in Piece since not all pieces have a batch
y.FOOID = FOOID 'Readonly from the Widget Class
y.FOONumber = FOONumber 'Readonly from the Widget class
y.update
```

Now for the big question. While MOST of the time we only need the FOOID and FOONumber as readonly data, there are occasions where someone would need

Architecting Dilemma

more information such as the chemistry and the weight (there will be some cases where weight would be very necessary), should we create two classes for FOO? One class would be created when we are creating a FOO for the first time and entered into the system, the other would be readonly data that a function in Widget would create. Example:

```
Public Class FOO
  _Weight as Long
  _ID as integer
  _PartID as string
```

Properties here

Methods here

```
Public Class FOO2
  _Weight as Long
  _ID as integer
  _PartID as string
  Readonly Properties here
  No methods.
```

Then call the class properties as follows:

```
Public Class Widget
  'Widget Properties
  Dim _X as new FOO2 ([ID])
  Public Readonly Property FOOWeight as long
  Return _x.Weight
```

Or should I create one FOO class and set readonly properties for FOO in my Widget Class and set them as appropriate.

```
Public Class Widget
  Dim _FOOID as Integer
  Dim _FOONumber as String
  Dim _FOOWeight as long
  Public Readonly Property _FOOWeight as Long
  return _FOOID
  ...More FOO Read only properties
  ...Widget Properties
  ...Widget Events
  ...Widget Methods
```

I know this is a lot but I would like some direction.

Thanks.

John Wright

