How to reduce the number of Set methods in a component

QUESTION

How to reduce the number of methods in a component I have the following class:

```
type TMyClass = class
  protected
    FProp1: string;
    bIsChanged: Boolean;
  public
    property Prop1: string read FProp1 write SetProp1;
end;
```

The *Prop1* property is simply for storing a value, it does no work on the class itself. The reason I have *SetProp1* is purely to update the *bIsChanged* variable. When the class is instantiated *bIsChanged* is defaulted to False. When *Prop1* is set *bIsChanged* is updated to true. This works fine as it is but sometimes I have like 20 properties in the class and for each one I have to create a *Set{Propery Name}* method so that *bIsChanged* can be set to true. What I would like to know is is there a way to have the class "notified" when any of it's properties are modified. This way I can declare my properties as:

```
property Prop1: string read FProp1 write FProp1;
```

and eliminate all the *Set{Property Name}* methods. When the class is "notified" *bIsChanged* is set to true and viola - I am set. I know I can create a *IsChanged* property and expose it to the user of the class and then it is up to them to set it but I would like to not do this.

There is no way to avoid having a *SetProp* method, however the number of methods can be reduced by sharing a single *SetProp* method for all properties of a single type. The secret is to use the Indexed Property feature. The following is extracted from my *TVersionInfo* component:

```
{ ... }
  private
    procedure SetVerProp(index: integer; value: TControl);
    function GetVerProp(index: integer): TControl;
  published
    property CtlCompanyName: TControl index 1
      read GetVerProp write SetVerProp;
    property CtlFileDescription: TControl index 2
      read GetVerProp write SetVerProp;
    property CtlFileVersion: TControl index 3
      read GetVerProp write SetVerProp;
    property CtlInternalName: TControl index 4
      read GetVerProp write SetVerProp;
    property CtlLegalCopyRight: TControl index 5
      read GetVerProp write SetVerProp;
    property CtlOriginalFileName: TControl index 6
      read GetVerProp write SetVerProp;
    property CtlProductName: TControl index 7
      read GetVerProp write SetVerProp;
    property CtlProductVersion: TControl index 8
      read GetVerProp write SetVerProp;
  end;
```

In your case you do not need the *GetXXX* method though it could also be used. The *SetXXX* method uses a simple case statement to set the right member:

```
procedure TVersionInfo.SetVerProp(index: integer; value: TControl);
begin
  case index of
    1: FCtlCompanyName := Value;
    2: FCtlFileDescription := Value;
    3: FCtlFileVersion := Value;
    4: FCtlInternalName := Value;
    5: FCtlLegalCopyRight := Value;
```

```
    6: FCtlOriginalFileName := Value;
    7: FCtlProductName := Value;
    8: FCtlProductVersion := Value;
  end;
  Refresh;
end;
```

In your case, you can replace *Refresh* with setting the *bIsChanged* value.

Original resource:	The Delphi Pool
Author:	Wayne Niddery
Added:	2009-10-26
Last updated:	2009-10-26