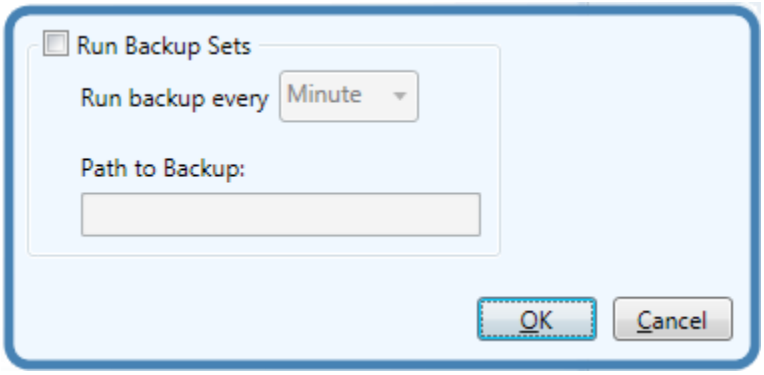
WPF: CheckBox as GroupBox Header

August 7th, 2009 by Mel Leave a reply »

This is a followup post to the one I wrote on Enabling Controls with a CheckBox. In that post we created some XAML that would enable/disable controls in the GUI based on the IsCheckedproperty of a checkbox.
Here's an enhancement to that:



So what we're doing is giving the controls on our dialog a nice visual grouping letting the user know that they're associated. The ***GroupBox*** *element has long existed to fulfill this need. However, with this little XAML change it also doubles as a sort of access control for the items it contains.*
Here's the XAML:

```
<GroupBox Padding="5" HorizontalAlignment="Stretch">
    <GroupBox.Header>
        <CheckBox x:Name="chkEnableBackup">Run Backup Sets</CheckBox>
    </GroupBox.Header>

    <StackPanel>
        <StackPanel Orientation="Horizontal">
            <Label Margin="12,0,0,0">Run backup every</Label>
            <ComboBox Width="70" SelectedIndex="0"
            IsEnabled="{Binding ElementName=chkEnableBackup, Path=IsChecked}">
                <ComboBoxItem>Minute</ComboBoxItem>
                <ComboBoxItem>Hour</ComboBoxItem>
                <ComboBoxItem>Day</ComboBoxItem>
            </ComboBox>
        </StackPanel>
        <StackPanel Margin="12,10,0,0">
            <Label>Path to Backup:</Label>
            <TextBox Width="200" Margin="5,0,0,0"
            IsEnabled="{Binding ElementName=chkEnableBackup, Path=IsChecked}"/>
        </StackPanel>
    </StackPanel>
</GroupBox>
```

So all we've done here is add the ***CheckBox*** *to the* ***<GroupBox.Header>*** *element of the****GroupBox****. Pretty slick!*
This little technique is complete UI Candy and in my opinion illustrates one of the many powerful features of WPF: the ability to customize the GUI in any way you want, down to any level!