

IT 认证电子书



质 量 更 高 服 务 更 好

半年免费升级服务

<http://www.itrenzheng.com>

Exam : **070-506 VB**

Title : TS: Microsoft Silverlight 4,
Development

Version : Demo

1. You are developing a Silverlight 4 application.

The application defines the following three event handlers. (Line numbers are included for reference only.)

```

01 Private Sub HandleCheck(sender As Object, e As RoutedEventArgs)
02     MessageBox.Show("Checked")
03 End Sub
04
05 Private Sub HandleUnchecked(sender As Object, e As RoutedEventArgs)
06     MessageBox.Show("Unchecked")
07 End Sub
08
09 Private Sub HandleThirdState(sender As Object, e As RoutedEventArgs)
10     MessageBox.Show("Indeterminate")
11 End Sub

```

You need to allow a check box that can be selected, cleared, or set to Indeterminate. You also need to ensure that the event handlers are invoked when the user changes the state of the control.

Which XAML fragment should you use?

- A. `<CheckBox x:Name="cb2" Content="Three State CheckBox" IsChecked="True" Checked="HandleCheck" Indeterminate="HandleUnchecked" Unchecked="HandleUnchecked" />`
- B. `<CheckBox x:Name="cb2" Content="Three State CheckBox" IsThreeState="True" Checked="HandleCheck" Indeterminate="HandleThirdState" Unchecked="HandleUnchecked" />`
- C. `<CheckBox x:Name="cb2" Content="Three State CheckBox" IsHitTestVisible="True" Checked="HandleCheck" Indeterminate="HandleThirdState" Unchecked="HandleUnchecked" />`
- D. `<CheckBox x:Name="cb2" Content="Three State CheckBox" IsEnabled="True" Checked="HandleCheck" Indeterminate="HandleUnchecked" Unchecked="HandleUnchecked" />`

Answer: B

2. You are developing a Silverlight 4 application.

The application contains an XAML page that defines the following Grid control.

```

<Grid Name="gridBody" >
    <Grid.RowDefinitions>
        <RowDefinition />
        <RowDefinition />
    </Grid.RowDefinitions>
    <TextBlock Text="Employee Info" />
    <TextBlock Text="Please enter employee info" Grid.Row="1" Height="20" VerticalAlignment="Top" />
    <TextBox x:Name="EmpInfo" Grid.Row="1" Margin="0,25,0,0" TextWrapping="Wrap" />
    ...
</Grid>

```

The code-behind file for myPage.xaml contains the following code segment. (Line numbers are included for reference only.)

```
01 Public Sub New()  
02     InitializeComponent()  
03  
04     Dim control As UserControl = New MyCustomControl()  
05  
06 End Sub
```

You need to replace the contents of the second row of gridBody with a user control of the MyCustomControl type.

Which code segment should you insert at line 05?

- A. `gridBody.Children.Insert(1, control)`
- B. `gridBody.RowDefinitions.Remove(gridBody.RowDefinitions(1))`
`gridBody.Children.Insert(1, control)`
- C. `gridBody.RowDefinitions.Remove(gridBody.RowDefinitions(1))`
`gridBody.Children.Insert(1, control)`
`gridBody.Children.Clear()`
`Grid.SetRow(control, 1)`
`gridBody.Children.Add(control)`
- D. `gridBody.RowDefinitions.Remove(gridBody.RowDefinitions(1))`
`gridBody.Children.Insert(1, control)`
`gridBody.Children.Clear()`
`Grid.SetRow(control, 1)`
`gridBody.Children.Add(control)`

Answer: D

3. You are developing a Silverlight 4 application.

The application defines the following XAML fragment. (Line numbers are included for reference only.)

```
01 <ComboBox>  
02     <ComboBoxItem Content="Item 1" />  
03     <ComboBoxItem Content="Item 2" />  
04     <ComboBoxItem Content="Item 3" />  
05 </ComboBox>
```

The code-behind file contains the following code segment. (Line numbers are included for reference only.)

```
06 Private Sub PrintText(sender As Object, args As SelectionChangedEventArgs)  
07  
08     MessageBox.Show("You selected " + cbi.Content.ToString() + ".")  
09 End Sub
```

You need to ensure that when the user selects an item in a ComboBox control, the content of the item is displayed.

What should you do?

A. Replace the following XAML fragment at line 01.

```
<ComboBox SelectionChanged="PrintText">
```

Add the following code segment at line 07.

```
Dim cbi As ComboBoxItem = TryCast(TryCast(sender, ComboBox).SelectedItem, ComboBoxItem)
```

B. Replace the following XAML fragment at line 01.

<ComboBox SelectionChanged="PrintText">

Add the following code segment at line 07.

```
Dim cbi As ComboBoxItem = TryCast(TryCast(sender, ComboBox).SelectedItem, ComboBoxItem)
```

C. Replace the following XAML fragment at line 01.

<ComboBox DropDownClosed="PrintText">

Add the following code segment at line 07.

```
Dim cbi As ComboBoxItem = TryCast(TryCast(sender, ComboBox).SelectedItem, ComboBoxItem)
```

D. Replace the following XAML fragment at line 01.

<ComboBox DropDownClosed="PrintText">

Add the following code segment at line 07.

```
Dim cbi As ComboBoxItem = TryCast(TryCast(sender, ComboBox).SelectedItem, ComboBoxItem)
```

Answer: A

4. You are developing a Silverlight 4 application.

You have a collection named ColPeople of the List<Person> type.

You define the Person class according to the following code segment.

```
Public Class Person
```

```
    Public Property Name() As String
```

```
    Public Property Description() As String
```

```
    Public Property Gender() As String
```

```
    Public Property Age() As Integer
```

```
    Public Property Weight() as Integer
```

```
End Class
```

You need to bind ColPeople to a ComboBox so that only the Name property is displayed.

Which XAML fragment should you use?

A. <ComboBox DataContext="{Binding ColPeople}" ItemsSource="{Binding ColPeople}" DisplayMemberPath="Name" />

B. <ComboBox DataContext="{Binding Person}" ItemsSource="{Binding Person}" DisplayMemberPath="ColPeople" />

C. <ComboBox DataContext="{Binding ColPeople}" DisplayMemberPath="Name" />

D. <ComboBox DataContext="{Binding Person}" />

Answer: A

5. You are developing a Silverlight 4 application.

You define an Invoice object according to the following code segment.

```
Public Class Invoice
```

```
    Public Property InvoiceId() As Integer
```

```
    Public Property Amount() As Double
```

```
    Public Property Supplier() As Supplier
```

```
    Public Property InvoiceDate() As DateTime
```

```
    Public Property PayDate() As DateTime
```

```
    Public Property InvoiceDescription() As String
```

```
End Class
```

You need to display a list of invoices that have the following properties displayed on each line: InvoiceId,

Amount, and InvoiceDate.

Which XAML fragment should you use?

- A. `<ListBox x:Name="InvoiceListBox">`
 `<StackPanel Orientation="Horizontal">`
 `<TextBlock Text="{Binding Path=InvoiceId}" />`
 `<TextBlock Text="{Binding Path=Amount}" />`
 `<TextBlock Text="{Binding Path=InvoiceDate}" />`
 `</StackPanel>`
`</ListBox>`
- B. `<ListBox x:Name="InvoiceListBox">`
 `<StackPanel Orientation="Horizontal">`
 `<ListBoxItem>`
 `<TextBlock Text="{Binding Path=InvoiceId}" />`
 `</ListBoxItem>`
 `<ListBoxItem>`
 `<TextBlock Text="{Binding Path=Amount}" />`
 `</ListBoxItem>`
 `<ListBoxItem>`
 `<TextBlock Text="{Binding Path=InvoiceDate}" />`
 `</ListBoxItem>`
 `</StackPanel>`
`</ListBox>`
- C. `<ListBox x:Name="InvoiceListBox">`
 `<ListBox.Items>`
 `<ItemsPanelTemplate>`
 `<StackPanel Orientation="Horizontal">`
 `<TextBlock Text="{Binding Path=InvoiceId}" />`
 `<TextBlock Text="{Binding Path=Amount}" />`
 `<TextBlock Text="{Binding Path=InvoiceDate}" />`
 `</StackPanel>`
 `</ItemsPanelTemplate>`
 `</ListBox.Items>`
`</ListBox>`
- D. `<ListBox x:Name="InvoiceListBox">`
 `<ListBox.ItemTemplate>`
 `<DataTemplate>`
 `<StackPanel Orientation="Horizontal">`
 `<TextBlock Text="{Binding Path=InvoiceId}" />`
 `<TextBlock Text="{Binding Path=Amount}" />`
 `<TextBlock Text="{Binding Path=InvoiceDate}" />`
 `</StackPanel>`
 `</DataTemplate>`
 `</ListBox.ItemTemplate>`
`</ListBox>`

Answer: D

6. You are developing a Silverlight 4 application.

You define the visual behavior of a custom control in the ControlTemplate by defining a VisualState object named Selected.

You need to change the visual state of the custom control to the Selected state.

Which code segment or XAML fragment should you use?

A. VisualStateManager.GoToState(Me, "Selected", True)

B. <VisualTransition To="Selected">

<Storyboard>

...

</Storyboard>

</VisualTransition>

C. <VisualTransition From="Selected">

<Storyboard>

...

</Storyboard>

</VisualTransition>

D. Public Shared ReadOnly SelectedProperty As DependencyProperty =

DependencyProperty.Register("Selected", GetType(VisualState), GetType(MyControl), Nothing)

Public Property Selected As VisualState

Get

Return GetValue(SelectedProperty)

End Get

Set(ByVal value As VisualState)

SetValue(SelectedProperty, value)

End Set

End Property

Answer: A

7. You are developing an application by using Silverlight 4 and Microsoft .NET Framework 4.

You create a new user control in the application. You add the following XAML fragment to the control.

```
<StackPanel KeyDown="App_KeyDown" Orientation="Vertical">
```

```
<TextBox x:Name="firstName" />
```

```
<TextBox x:Name="lastName" />
```

```
<TextBox x:Name="address" />
```

```
</StackPanel>
```

You add the following code segment in the code-behind file of the control. (Line numbers are included for reference only.)

```
01 Private Sub App_KeyDown(ByVal sender As Object, ByVal e As KeyEventArgs)
```

```
02
```

```
03 End Sub
```

```
04
```

```
05 Private Sub FirstAndLastNameKeyDown()
```

06 '...

07 End Sub

You need to ensure that the FirstAndLastNameKeyDown method is invoked when a key is pressed while the focus is on the firstName or lastName TextBox controls. You also need to ensure that the default behavior of the controls remains unchanged.

Which code segment should you add at line 02?

A. If ((CType(sender,FrameworkElement).Name = "firstName") _
OrElse (CType(sender,FrameworkElement).Name = "lastName")) Then
FirstAndLastNameKeyDown

End If

e.Handled = false

B. If ((CType(sender,FrameworkElement).Name = "firstName") _
OrElse (CType(sender,FrameworkElement).Name = "lastName")) Then
FirstAndLastNameKeyDown

End If

e.Handled = true

C. If ((CType(e.OriginalSource,FrameworkElement).Name = "firstName") _
OrElse (CType(e.OriginalSource,FrameworkElement).Name = "lastName")) Then
FirstAndLastNameKeyDown

End If

e.Handled = false

D. If ((CType(e.OriginalSource,FrameworkElement).Name = "firstName") _
OrElse (CType(e.OriginalSource,FrameworkElement).Name = "lastName")) Then
FirstAndLastNameKeyDown

End If

e.Handled = true

Answer: C

8. You are developing an application by using Silverlight 4 and Microsoft .NET Framework 4. The application has a TextBox control named txtName.

You need to handle the event when txtName has the focus and the user presses the F2 key.

Which two actions should you perform? (Each correct answer presents part of the solution. Choose two.)

A. AddHandler txtName.KeyDown, New KeyEventHandler (AddressOf txtName_KeyDown)

B. AddHandler txtName.LostFocus, New RoutedEventHandler(AddressOf txtName_LostFocus)

C. AddHandler txtName.TextChanged, New TextChangedEventHandler(AddressOf
txtName_TextChanged)

D. Private Sub txtName_TextChanged(sender As Object, e As TextChangedEventArgs)

'Custom logic

If DirectCast(e.OriginalSource, Key) = Key.F2 Then

End If

End Sub

E. Private Sub txtName_KeyDown(sender As Object, e As KeyEventArgs)

'Custom logic


```

    If e.Key = Key.F2 Then
    End If
End Sub
F. Private Sub txtName_LostFocus(sender As Object, e As RoutedEventArgs)
    'Custom logic
    If DirectCast(e.OriginalSource, Key) = Key.F2 Then
    End If
End Sub
Answer: AE

```

9. You are developing an application by using Silverlight 4 and Microsoft .NET Framework 4.

The application contains the following XAML fragment.

```
<TextBlock x:Name="QuoteOfTheDay" />
```

The application calls a Windows Communication Foundation (WCF) service named MyService that returns the quote of the day and assigns it to the QuoteOfTheDay TextBlock.

The application contains the following code segment. (Line numbers are included for reference only.)

```

01 Dim client = New MyService.MyServiceClient()
02 AddHandler client.GetQuoteOfTheDayCompleted, Sub(s, args) QuoteOfTheDay.Text = args.Result
03 client.GetQuoteOfTheDayAsync()

```

You need to handle errors that might occur as a result of the service call. You also need to provide a default value of "Unavailable" when an error occurs.

Which code segment should you replace at lines 02 and 03?

A. QuoteOfTheDay.Text = "Unavailable"

```
AddHandler client.GetQuoteOfTheDayCompleted, Sub(s, args)
```

```
    QuoteOfTheDay.Text = args.Result
```

```
End Sub
```

```
client.GetQuoteOfTheDayAsync()
```

B. AddHandler client.GetQuoteOfTheDayCompleted,Sub(s, args)

```
    If args.Result IsNot Nothing Then
```

```
        QuoteOfTheDay.Text = args.Result
```

```
    Else
```

```
        QuoteOfTheDay.Text = "Unavailable"
```

```
    End If
```

```
End Sub
```

```
client.GetQuoteOfTheDayAsync()
```

C. AddHandler client.GetQuoteOfTheDayCompleted, Sub(s, args)

```
    QuoteOfTheDay.Text = args.Result
```

```
End Sub
```

```
Try
```

```
    client.GetQuoteOfTheDayAsync()
```

```
Catch ex As Exception
```

```
    ' TODO: handle exception
```

```
    QuoteOfTheDay.Text = "Unavailable"
```

```
End Try
```

D. AddHandler client.GetQuoteOfTheDayCompleted, Sub(s, args)

```
If args.[Error] Is Nothing Then
    QuoteOfTheDay.Text = args.Result
Else
    ' TODO: handle error
    QuoteOfTheDay.Text = "Unavailable"
End If
```

End Sub

client.GetQuoteOfTheDayAsync()

Answer: D

10. You are developing an application by using Silverlight 4 and Microsoft .NET Framework 4. You create a Windows Communication Foundation (WCF) Data Service. You add a service reference to the WCF Data Service named NorthwindEntities in the Silverlight application. You also add a CollectionViewSource object named ordersViewSource in the Silverlight application. You add the following code segment. (Line numbers are included for reference only.)

```
01 Sub getOrders_Click(ByVal sender As Object, ByVal e As RoutedEventArgs)
02     Dim context As New NorthwindEntities()
03
04     Dim query = From order In context.Orders Select order
05
06 End Sub
```

You need to retrieve the Orders data from the WCF Data Service and bind the data to the ordersViewSource object.

Which two actions should you perform? (Each correct answer presents part of the solution. Choose two.)

A. Add the following code segment at line 03.

```
Dim obsCollection = New ObservableCollection(Of Order)
```

B. Add the following code segment at line 03.

```
Dim dsOrders As New DataServiceCollection(Of Order)
```

```
AddHandler dsOrders.LoadCompleted, New EventHandler(Of LoadCompletedEventArgs)(Sub(dsc, args)
ordersViewSource.Source = dsOrders
End Sub)
```

C. Add the following code segment at line 05.

```
dsOrders.LoadAsync(query)
```

D. Add the following code segment at line 05.

```
dsOrders.Load(query)
```

E. Add the following code segment at line 05.

```
query.ToList().ForEach(Sub(o) obsCollection.Add(o))
ordersViewSource.Source = obsCollection
```

Answer: BC

11. You are developing an application by using Silverlight 4 and Microsoft .NET Framework 4.

You add a BackgroundWorker object named worker to the application.

You add the following code segment. (Line numbers are included for reference only.)

```
01 Public Sub New()  
02     InitializeComponent()  
03     worker.WorkerSupportsCancellation = True  
04     AddHandler worker.DoWork, New DoWorkEventHandler(AddressOf worker_DoWork)  
05     AddHandler worker.RunWorkerCompleted, New RunWorkerCompletedEventHandler(AddressOf  
worker_Completed)  
06 End Sub  
07 Private Sub worker_DoWork(sender As Object, e As DoWorkEventArgs)  
08     For i As Integer = 0 To 99  
09         InvokeLongRunningProcessStep()  
10     Next  
11 End Sub
```

You need to ensure that worker can be properly canceled.

Which code segment should you use to replace line 09?

A. Dim cancel = TryCast(sender, BackgroundWorker).CancellationPending

If cancel Then

TryCast(sender, BackgroundWorker).CancelAsync()

Else

InvokeLongRunningProcessStep()

End If

B. Dim cancel = TryCast(sender, BackgroundWorker).CancellationPending

If cancel Then

e.Cancel = True

Else

InvokeLongRunningProcessStep()

End If

C. Dim cancel = e.Cancel

If cancel Then

TryCast(sender, BackgroundWorker).CancelAsync()

Else

InvokeLongRunningProcessStep()

End If

D. Dim cancel = e.Cancel

If cancel Then

e.Cancel = True

Else

InvokeLongRunningProcessStep()

End If

Answer: B

12. You are developing an application by using Silverlight 4 and Microsoft .NET Framework 4.

You add a BackgroundWorker object named worker to the application. You also add a CheckBox control named checkBox and a TextBlock control named statusTextBlock.

You add the following code segment. (Line numbers are included for reference only.)

```
01 Public Sub New()  
02     InitializeComponent()  
03     worker.WorkerReportsProgress = True  
04     AddHandler worker.DoWork, New DoWorkEventHandler(AddressOf worker_DoWork)  
05     AddHandler worker.ProgressChanged, New ProgressChangedEventArgs(AddressOf  
worker_ProgressChanged)  
06 End Sub  
07 Private Sub worker_DoWork(sender As Object, e As DoWorkEventArgs)  
08     For i As Integer = 0 To 99  
09         Dim isChecked As Boolean = checkBox.IsChecked.HasValue AndAlso  
checkBox.IsChecked.Value  
10         ExecuteLongRunningProcessStep(isChecked)  
11         worker.ReportProgress(i)  
12     Next  
13 End Sub  
14 Private Sub worker_ProgressChanged(sender As Object, e As ProgressChangedEventArgs)  
15     statusTextBlock.Text = Convert.ToString(e.ProgressPercentage) & "%"  
16 End Sub
```

You attempt to run the application. You receive the following error message:

"Invalid cross-thread access."

You need to ensure that worker executes successfully.

What should you do?

A. Replace line 09 with the following code segment.

```
Dim b = CType(checkBox.GetValue(CheckBox.IsCheckedProperty), System.Nullable(Of Boolean))  
Dim isChecked As Boolean = b.HasValue AndAlso b.Value
```

B. Replace line 09 with the following code segment.

```
Dim isChecked As Boolean = False  
Dispatcher.BeginInvoke(Function()  
isChecked = checkBox.IsChecked.HasValue AndAlso checkBox.IsChecked.Value  
End Function)
```

C. Replace line 15 with the following code segment.

```
statusTextBlock.SetValue(TextBlock.TextProperty, (e.ProgressPercentage + "%"))
```

D. Replace line 15 with the following code segment.

```
Dispatcher.BeginInvoke(Function()  
statusTextBlock.Text = e.ProgressPercentage + "%"  
End Function)
```

Answer: B

13. You are developing an application by using Silverlight 4 and Microsoft .NET Framework 4.

You add the following code segment. (Line numbers are included for reference only.)

```
01 Public Class MyControl Inherits Control  
02  
03     Public Property Title() As String
```

```
04     Get
05         Return DirectCast(GetValue(TitleProperty), String)
06     End Get
07     Set
08         SetValue(TitleProperty, value)
09     End Set
10 End Property
11 End Class
```

You need to create a dependency property named TitleProperty that allows developers to set the Title. You also need to ensure that the default value of the TitleProperty dependency property is set to Untitled. Which code segment you add at line 02?

- A. `Public Shared ReadOnly TitleProperty As DependencyProperty = DependencyProperty.Register("Untitled", GetType(System.String), GetType(MyControl), Nothing)`
- B. `Public Shared ReadOnly TitleProperty As DependencyProperty = DependencyProperty.Register("Untitled", GetType(System.String), GetType(MyControl), New PropertyMetadata("Title"))`
- C. `Public Shared ReadOnly TitleProperty As DependencyProperty = DependencyProperty.Register("Title", GetType(System.String), GetType(MyControl), New PropertyMetadata("Untitled"))`
- D. `Public Shared ReadOnly TitleProperty As DependencyProperty = DependencyProperty.Register("Title", GetType(String), GetType(MyControl), New PropertyMetadata(New PropertyChangedCallback(Sub(depObj, args) depObj.SetValue(MyControl.TitleProperty, "Untitled") End Sub)))`

Answer: C

14. You are developing an application by using Silverlight 4 and Microsoft .NET Framework 4. You create a control named MyControl in the application. Each instance of the control contains a list of FrameworkElement objects.

You add the following code segment. (Line numbers are included for reference only.)

```
01 Public Class MyControl
02     Inherits Control
03
04     Public ReadOnly Property ChildElements As List(Of FrameworkElement)
05     Get
06         Return DirectCast(GetValue(MyControl.ChildElementsProperty), List(Of FrameworkElement))
07     End Get
08 End Property
09 Public Sub New()
10
11 End Sub
12
13 Shared Sub New()
```

14

15 End Sub

16 End Class

You need to create the ChildElementsProperty dependency property. You also need to initialize the property by using an empty list of FrameworkElement objects.

Which two actions should you perform? (Each correct answer presents part of the solution. Choose two.)

A. Add the following code segment at line 03.

```
Public Shared ReadOnly ChildElementsProperty As DependencyProperty =  
    DependencyProperty.Register("ChildElements", GetType(List(Of  
        FrameworkElement)), GetType(MyControl), New PropertyMetadata(New  
        List(Of FrameworkElement)()))
```

B. Add the following code segment at line 03.

```
Public Shared ReadOnly ChildElementsProperty As DependencyProperty =  
    DependencyProperty.Register("ChildElements", GetType(List(Of  
        FrameworkElement)), GetType(MyControl), New  
        PropertyMetadata(Nothing))
```

C. Add the following code segment at line 10.

```
SetValue(MyControl.ChildElementsProperty, New List(Of  
    FrameworkElement)())
```

D. Add the following code segment at line 14.

```
ChildElementsProperty = DependencyProperty.Register("ChildElements",  
    GetType(List(Of FrameworkElement)), GetType(MyControl), New  
    PropertyMetadata(New List(Of FrameworkElement)()))
```

Answer: BC

15. You are developing an application by using Silverlight 4 and Microsoft .NET Framework 4.

You add the following code segment. (Line numbers are included for reference only.)

```
01 Dim outerCanvas = New Canvas()  
02 Dim innerCanvas = New Canvas()  
03 innerCanvas.Width = 200  
04 innerCanvas.Height = 200  
05 outerCanvas.Children.Add(innerCanvas)  
06
```

You need to set the distance between the left of the innerCanvas element and the left of the outerCanvas element to 150 pixels.

Which code segment should you add at line 06?

A. `outerCanvas.Margin = New Thickness(0.0, 150.0, 0.0, 0.0)`

B. `innerCanvas.Margin = new Thickness(0.0, 150.0, 0.0, 0.0)`

C. `outerCanvas.SetValue(Canvas.LeftProperty, 150.0)`

D. `innerCanvas.SetValue(Canvas.LeftProperty, 150.0)`

Answer: D