



**70-502(CSharp)**

**Windows Presentation Foundation App Dev**

Q&A

DEMO Version

Copyright (c) 2009 Chinatag LLC. All rights reserved.

## **Important Note Please Read Carefully**

For demonstration purpose only, this free version Chinatag study guide contains **10** full length questions selected from our full version products which have more than **200** questions each.

This Study guide has been carefully written and compiled by Chinatag certification experts. It is designed to help you learn the concepts behind the questions rather than be a strict memorization tool. Repeated readings will increase your comprehension.

For promotion purposes, all PDF files are **not** encrypted. Feel free to distribute copies among your friends and let them know Chinatag website.

## **Study Tips**

This product will provide you questions and answers along with detailed explanations carefully compiled and written by our experts. Try to understand the concepts behind the questions instead of cramming the questions. Go through the entire document at least twice so that you make sure that you are not missing anything.

## **Latest Version**

We are constantly reviewing our products. New material is added and old material is revised. Free updates are available for 90 days after the purchase. You should check the products page on the <http://www.chinatag.com> website for an update 3-4 days before the scheduled exam date.

Please tell us what you think of our products. We appreciate both positive and critical comments as your feedback helps us improve future versions. Feedback on specific questions should be send to [feedback@chinatag.com](mailto:feedback@chinatag.com).

Thanks for purchasing our products and look forward to supplying you with all your Certification training needs.

Good studying!

Technical and Support Team  
Chinatag LLC.

**QUESTION 1**

You are creating a Windows Presentation Foundation application by using Microsoft .NET Framework 3.5. You create a window for the application. You need to ensure that the following requirements are met:

- An array of strings is displayed by using a ListBox control in a two-column format.
- The data in the ListBox control flows from left to right and from top to bottom.

Which ListBox control would you use?

A. `<ListBox Name="myList ">`  
`<ListBox.ItemsPanel>`  
`<ItemsPanelTemplate>`  
`<UniformGrid Columns="2" />`  
`</ItemsPanelTemplate>`  
`</ListBox.ItemsPanel>`  
`</ListBox>`

Use the following C# code to associate the array of strings to the ListBox control.

```
myList.ItemsSource = arrayOfString
```

B. `<ListBox Name="myList ">`  
`<ListBox.ItemsPanel>`  
`<ItemsPanelTemplate>`  
`<StackPanel />`  
`</ItemsPanelTemplate>`  
`</ListBox.ItemsPanel>`  
`</ListBox>`

Use the following C# code to associate the array of strings to the ListBox control.

```
myList.ItemsSource = arrayOfString
```

C. `<ListBox Name="myList ">`  
`<ListBox.ItemsPanel>`  
`<ItemsPanelTemplate>`  
`<WrapPanel />`  
`</ItemsPanelTemplate>`  
`</ListBox.ItemsPanel>`  
`</ListBox>`

Use the following C# code to associate the array of strings to the ListBox control.

```
myListView.ItemsSource = arrayOfString
```

D. `<ListBox Name="myList ">`  
`<ListBox.ItemsPanel>`  
`<ItemsPanelTemplate>`  
`<Grid>`  
`<Grid.ColumnDefinitions>`  
`<ColumnDefinition />`  
`<ColumnDefinition />`  
`</Grid.ColumnDefinitions>`  
`</Grid>`  
`</ItemsPanelTemplate>`  
`</ListBox.ItemsPanel>`  
`</ListBox>`

Use the following C# code to associate the array of strings to the ListBox control.

```
myList.ItemsSource = arrayOfString
```

**Answer: A**

**QUESTION 2**

You create a form by using Windows Presentation Foundation and Microsoft .NET Framework 3.5. The form contains a status bar.

You plan to add a ProgressBar control to the status bar. You need to ensure that the ProgressBar control displays the progress of a task for which you cannot predict the completion time.

Which code segment should you use?

- A. progbar.IsIndeterminate = true
- B. progbar.IsIndeterminate = false
- C. progbar.HasAnimatedProperties = true
- D. progbar.HasAnimatedProperties = false

**Answer: A**

### QUESTION 3

You are converting a Windows Forms application to a Windows Presentation Foundation (WPF) application.

You use Microsoft .NET Framework 3.5 to create the WPF application.

The WPF application will reuse 30 forms of the Windows Forms application. The WPF application contains the following class definition.

```
public class OwnerWindow : System.Windows.Forms.IWin32Window
{
    private IntPtr handle_Renamed;
    public IntPtr Handle
    {
        get { return handle_Renamed ; }
        set { handle_Renamed = value ; }
    }
}
```

You write the following code segment in the WPF application.

(Line numbers are included for reference only.)

```
01 public System.Windows.Forms.DialogResult LaunchWindowsFormsDialog(System.
Windows.Forms.Form dialog, Window wpfParent)
02 {
03     System.Windows.Interop.WindowInteropHelper helper = new System.Windows.
Interop.WindowInteropHelper(wpfParent) ;
04     OwnerWindow owner = new OwnerWindow() ;
05
06 }
```

You need to ensure that the application can launch the reusable forms as modal dialogs.

Which code segment should you insert at line 05?

- A. owner.Handle = helper.Owner;  
DialogResult db = new DialogResult();  
return db;
- B. owner.Handle = helper.Owner  
return dialog.ShowDialog(owner);
- C. owner.Handle = helper.Owner ;  
Nullable<bool> result = wpfParent.ShowDialog();  
if (result.HasValue)  
{  
 if (result.value)  
 return System.Windows.Forms.DialogResult.OK;  
 else  
 return System.Windows.Forms.DialogResult.Cancel;  
}  
else  
{  
 return System.Windows.Forms.DialogResult.Cancel;  
}

```

D. owner.Handle = helper.Handle:
  Nullable<bool> result = wpfParent.ShowDialog();
  if (result.HasValue)
  {
    if (result.value)
      return System.Windows.Forms.DialogResult.OK;
    else
      return System.Windows.Forms.DialogResult.Cancel;
  }
  else
  {
    return System.Windows.Forms.DialogResult.Cancel;
  }

```

**Answer: B**

#### QUESTION 4

You are creating a Windows Presentation Foundation application by using Microsoft .NET Framework 3.5. You create a window in the application. You plan to select a layout control to host the elements that you add to the window.

You need to select a control that meets the following requirements with its default properties:

1. The elements stretch horizontally to occupy the available width of the window.
2. The elements do not stretch vertically.

Which control should you use?

- A. The Grid control
- B. The Canvas control
- C. The WrapPanel control
- D. The StackPanel control

**Answer: D**

#### QUESTION 5

You are creating a Windows Presentation Foundation application by using Microsoft .NET Framework 3.5.

You need to perform the following tasks:

- Add a control to a window by using the following XAML code fragment.  
`<local:RedTextControl Background="Yellow" />`
- Ensure that the background color of the control is yellow.
- Ensure that the foreground color is red

What should you do?

- A. Add the following code segment to a code-behind file.
 

```

public class RedTextControl : ContentControl
{
    public void RedTextControl()
    {
        TextBlock tb = new TextBlock();
        tb.Foreground = Brushes.Red;
        tb.Text = "Hello";
        this.AddChild(tb);
    }
}

```

- B. Add the following code segment to a code-behind file.

```
public class RedTextControl : Control
{
    public void RedTextControl()
    {
        TextBlock tb = new TextBlock() ;
        tb.Foreground = Brushes.Red ;
        tb.Text = "Hello" ;
        this.AddLogicalChild(tb) ;
    }
}
```

- C. Add the following XAML code fragment to an XAML file.

```
<UserControl x:Class="RedTextControl" xmlns="..." xmlns:x="...">
  <DockPanel>
    <TextBlock Foreground="Red" Text="Hello" />
  </DockPanel>
</UserControl>
```

Add the following code segment to a code-behind file.

```
internal partial class RedTextControl
{
    public void RedTextControl()
    {
        InitializeComponent();
        Background.CoerceValue(TextBlock.BackgroundProperty);
    }
}
```

- D. Add the following XAML code fragment to an XAML file.

```
<UserControl x:Class="RedTextControl" xmlns="..." xmlns:x="...">
  <DockPanel>
    <TextBlock Foreground="Red" Text="Hello" />
  </DockPanel>
</UserControl>
```

Add the following code segment to a code-behind file.

```
internal partial class RedTextControl : UserControl
{
    public void RedTextControl()
    {
        InitializeComponent();
    }
}
```

**Answer: D**

### QUESTION 6

You are creating a Windows Presentation Foundation application by using Microsoft .NET Framework 3.5. You plan to add a Button control to a Canvas control. You need to ensure that exactly 10 device-independent pixels are present between the right side of the Button control and the right side of the Canvas control. Which XAML code fragment should you use?

- A. `<Canvas Margin="10">`  
`<Button>I'm a button</Button>`  
`</Canvas>`
- B. `<Canvas>`  
`<Button Canvas.Left="10" Canvas.Right="10">I'm a button</Button>`  
`</Canvas>`

- C. `<Canvas Margin="10">  
     <Button HorizontalAlignment="Right">I'm a button</Button>  
 </Canvas>`
- D. `<Canvas>  
     <Button Canvas.Top="10" Canvas.Right="10">I'm a button</Button>  
 </Canvas>`

**Answer: D**

### QUESTION 7

You are creating a Windows Presentation Foundation application by using Microsoft .NET Framework 3.5. You create a dialog window composed of a TextBox control and a Button control by using the following XAML code fragment.

```
<TextBox Width="200" AcceptsReturn="True" />
<Button Width="80" Click="Button_Click">OK</Button>
```

You need to ensure that each time the user presses the ENTER key, the click event of the OK button is raised. Which two actions should you perform? (Each correct answer presents part of the solution.)

- A. Add the `IsDefault="True"` attribute to the Button element.
- B. Add the `IsTabStop="True"` attribute to the Button element.
- C. Add the `IsHitTestVisible="True"` attribute to the Button element.
- D. Add the `Focusable="False"` attribute to the TextBox element.
- E. Add the `AcceptsTab="True"` attribute to the TextBox element.
- F. Set the `AcceptsReturn` attribute value of the TextBox element to False.

**Answer: AF**

### QUESTION 8

You are creating a Windows Presentation Foundation (WPF) application by using Microsoft .NET Framework 3.5.

You plan to implement a search functionality for a text editor. You write the following code segment for the WPF screen.

```
<DockPanel>
  <WrapPanel DockPanel.Dock="Top">
    <TextBox Name="tbxTextToFind" Width="200" />
    <Button Name="btnFind" Width="80" Click="btnFind_Click">Find</Button>
  </WrapPanel>
  <RichTextBox Name="rtbText" />
</DockPanel>
```

You need to ensure that on clicking the Find button, the value of the `tbxTextToFind` text box is selected in the `rtbText` control.

Which code segment should you use?

- A. 

```
FlowDocument doc = rtbText.Document;
string text = (new TextRange(doc.ContentStart, doc.ContentEnd)).Text;
int index = text.IndexOf(tbxTextToFind.Text);
TextPointer start = doc.ContentStart.GetPositionAtOffset(index);
TextPointer end = start.GetPositionAtOffset(tbxTextToFind.Text.Length);
rtbText.Selection.Select(start, end);
```

- B. `TextPointer cur = rtbText.Document.ContentStart;`  
`while (cur != null)`  
`{`  
`TextPointer end = cur.GetPositionAtOffset(tbxTextToFind.Text.Length);`  
`if (end != null )`  
`{`  
`TextRange search = new TextRange(cur, end);`  
`if (search.Text == tbxTextToFind.Text)`  
`{`  
`rtbText.Selection.Select(search.Start, search.End);`  
`break;`  
`}`  
`}`  
`cur = cur.GetNextContextPosition(LogicalDirection.Forward);`  
`}`
- C. `TextPointer cur = rtbText.Document.ContentStart;`  
`while (cur != null)`  
`{`  
`TextPointer end = cur.GetPositionAtOffset(tbxTextToFind.Text.Length);`  
`if (end != null )`  
`{`  
`TextRange search = new TextRange(cur, end);`  
`if (search.Text == tbxTextToFind.Text)`  
`{`  
`rtbText.Selection.Select(search.Start, search.End);`  
`break;`  
`}`  
`}`  
`cur = cur.GetNextInsertionPosition(LogicalDirection.Forward);`  
`}`
- D. `FlowDocument doc = rtbText.Document;`  
`string text = (new TextRange(doc.ContentStart, doc.ContentEnd)).Text;`  
`int index = text.LastIndexOf(tbxTextToFind.Text);`  
`TextPointer start = doc.ContentStart.GetPositionAtOffset(index + 1);`  
`TextPointer end = start.GetPositionAtOffset(tbxTextToFind.Text.Length);`  
`rtbText.Selection.Select(start, end);`

**Answer: C**

### QUESTION 9

You are creating a Windows Presentation Foundation application by using Microsoft .NET Framework 3.5.

You plan to add a Button control and a StatusBar control to a window.

You need to ensure that the two controls meet the following requirements:

- The StatusBar control sticks to the lower edge of the window.
- The Button control has the minimum required size to display its content.

Which XAML code fragment should you use?

- A. `<DockPanel>`  
`<StatusBar DockPanel.Dock="Bottom">Information</StatusBar>`  
`<Button>OK</Button>`  
`</DockPanel>`
- B. `<StackPanel>`  
`<Button>OK</Button>`  
`<StatusBar>Information</StatusBar>`  
`</StackPanel>`
- C. `<WrapPanel Orientation="Vertical">`  
`<Button>OK</Button>`  
`<StatusBar>Information</StatusBar>`  
`</WrapPanel>`

```
D. <DockPanel LastChildFill="False" >
    <WrapPanel DockPanel.Dock="Top" >
        <Button>OK</Button>
    </WrapPanel>
    <StatusBar DockPanel.Dock="Bottom">Information</StatusBar>
</DockPanel>
```

**Answer: D**

### QUESTION 10

You have created a Windows Presentation Foundation application by using Microsoft .NET Framework 3.5. The application, named EnterpriseApplication.exe, runs over the network.

You add the WindowSize parameter and the WindowPosition parameter to the Settings.settings file by using the designer at the User Scope Level.

The dimensions and position of the window are read from the user configuration file.

The application must retain the original window size and position for users executing the application.

You need to ensure that the following requirements are met:

- The window dimensions for each user are saved in the user configuration file.
- User settings persist when a user exits the application.

Which configuration setting should you use?

- A. 

```
private void OnClosing(object sender, System.ComponentModel.CancelEventArgs e)
{
    Properties.Settings.Default.WindowPosition = new Point(this.Left, this.Top);
    Properties.Settings.Default.WindowSize = new Size(this.Width, this.Height);
    Properties.Settings.Default.Save();
}
```
- B. 

```
private void OnClosing(object sender, System.ComponentModel.CancelEventArgs e)
{
    RegistryKey appKey = Registry.CurrentUser.CreateSubKey("Software\
EnterpriseApplication");
    RegistryKey settingsKey = appKey.CreateSubKey("WindowSettings");
    RegistryKey windowPositionKey = settingsKey.CreateSubKey("WindowPosition");
    RegistryKey windowSizeKey = settingsKey.CreateSubKey("WindowSize");
    windowPositionKey.SetValue("X", this.Left);
    windowPositionKey.SetValue("Y", this.Top);
    windowSizeKey.SetValue("Width", this.Width);
    windowSizeKey.SetValue("Height", this.Height);
}
```
- C. 

```
private void OnClosing(object sender, System.ComponentModel.CancelEventArgs e)
{
    System.Xml.XmlDocument doc = new System.Xml.XmlDocument();
    doc.Load("EnterpriseApplication.exe.config");
    System.Xml.XmlNode nodePosition = doc.SelectSingleNode("//setting
[@name='WindowPosition']");
    nodePosition.ChildNodes[0].InnerText = string.Format("{0},{1}", this.Left,
this.Top);
    System.Xml.XmlNode nodeSize = doc.SelectSingleNode("//setting
[@name='WindowSize']");
    nodeSize.ChildNodes[0].InnerText = string.Format("{0},{1}", this.Width, this.
Height);
    doc.Save("UserConfigDistractor2.exe.config");
}
```

```
D. private void Window_Closing(object sender, System.ComponentModel.  
CancelEventArgs e)  
{  
    StreamWriter sw = new StreamWriter("EnterpriseApplication.exe.config", true);  
    sw.WriteLine("<EnterpriseApplication.Properties.Settings>");  
    sw.WriteLine("<setting name=\"WindowSize\" serializeAs=\"String\">");  
    sw.WriteLine(string.Format("<value>{0},{1}</value>", this.Width, this.  
Height));  
    sw.WriteLine("</setting>");  
    sw.WriteLine("<setting name=\"WindowPosition\" serializeAs=\"String\">");  
    sw.WriteLine(string.Format("<value>{0},{1}</value>", this.Left, this.Top));  
    sw.WriteLine("</setting>");  
    sw.WriteLine("</UserConfigProblem.Properties.Settings>");  
    sw.Close();  
}
```

**Answer: A**