Passcert

Higher Quality, better service!

Q&A

Http://www.passcert.com

We offer free update service for one year.

Exam : **70-502 CSharpChinese**

Title : TS: Microsoft .NET Framework 3.5 – Windows Presentation Foundation

Version : Demo

1. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。您向 Window 元素中添加一个 CommandBinding 元素。该命令具有一个键盘笔势 Ctrl+H。Window 元素中包含下面的 MenuItem 控件。

<MenuItem Header="Highlight Content"

Command="local:CustomCommands.Highlight" />

您需要确保当焦点切换到不包含任何文本的 TextBox 控件时，MenuItem 控件处于禁用状态而且该命令不可执行。

您应该怎么办？

A. 在 TextBox 控件的 GotFocus 事件处理程序中，设置 MenuItem 控件的 IsEnabled 属性。

B. 将该命令的 CanExecute 属性设置为 Highlight_CanExecute。

将以下方法添加到该窗口的代码隐藏文件中。

```
private void Highlight_CanExecute(object sender, CanExecuteEventArgs e) {
   TextBox txtBox = sender as TextBox;
   e.CanExecute = (txtBox.Text.Length > 0);
}
```

C. 将该命令的 CanExecute 属性设置为 Highlight_CanExecute。

将以下方法添加到该窗口的代码隐藏文件中。

```
private void Highlight_CanExecute(object sender, CanExecuteEventArgs e) {
   TextBox txtBox = e.Source as TextBox;
   e.CanExecute = (txtBox.Text.Length > 0);
}
```

D. 将该命令的 CanExecute 属性设置为 Highlight_CanExecute。

将以下方法添加到该窗口的代码隐藏文件中。

```
private void Highlight_CanExecute(object sender, CanExecuteEventArgs e) {
   MenuItem menu = e.Source as MenuItem;
   TextBox txtBox = menu.CommandTarget as TextBox;
   Menu.IsEnabled = (txtBox.Text.Length > 0);
}
```

Answer: C

2. 您使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。

该应用程序名为 EnterpriseApplication.exe。

您使用设计器在用户作用域级别向 Settings.settings 文件中添加 WindowSize 参数和 WindowPosition 参数。窗口的大小和位置是从用户配置文件读取的。

该应用程序必须针对每个执行它的用户保持最初的窗口大小和位置。

您需要确保满足以下要求：

È 每个用户的窗口大小都保存在用户配置文件中。

È 在用户退出应用程序后保留用户设置。

您应该使用哪个配置设置？

A. private void OnClosing(object sender,
 System.ComponentModel.CancelEventArgs e){
  Settings.Default.WindowPosition = new Point (this.Left,
   this.Top);
  Settings.Default.WindowSize = new Size (this.Width,
   this.Height);
  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){
XmlDocument doc = new XmlDocument();
doc.Load("EnterpriseApplication.exe.config");
XmlNode nodePosition =
doc.SelectSingleNode("//setting[@name=\'WindowPosition\']");
nodePosition.ChildNodes[0].InnerText = String.Format("{0},{1}",
this.Left, this.Top);
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

3. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。该应用程序定义一个 BrowserWindow 类。BrowserWindow 类的每个实例允许用户在一个单独的窗口中浏览网站。在打开新的浏览器窗口时，用户将重定向到一个预定义的 URL。

您编写以下代码段。

```
01 private void OpenNewWindow(object sender, RoutedEventArgs e)
02 {
03     Thread newWindowThread = new Thread(new
        ThreadStart(NewThreadProc));
04
05     newWindowThread.Start();
06 }
07 private void NewThreadProc()
08 {
09
10   }
```

您需要确保满足以下要求：

È 在创建其他浏览器窗口时，应用程序的主窗口不被阻止。

È 在应用程序的主窗口关闭时，应用程序即执行完毕。

您应该怎么办？

A. 在第 04 行处插入以下代码段。

```
newWindowThread.SetApartmentState(ApartmentState.STA);
newWindowThread.IsBackground = true;
```

在第 09 行处插入以下代码段。

```
BrowserWindow newWindow = new BrowserWindow();
newWindow.Show();
Application app = new Application();
```

app.Run(newWindow);

B. 在第 04 行处插入以下代码段。

newWindowThread.IsBackground = true;

在第 09 行处插入以下代码段。

newWindowThread.SetApartmentState(ApartmentState.STA);

BrowserWindow newWindow = new BrowserWindow();

newWindow.Show();

Application app = new Application();

app.Run(newWindow);

C. 在第 04 行处插入以下代码段。

newWindowThread.SetApartmentState(ApartmentState.STA);

newWindowThread.IsBackground = false;

在第 09 行处插入以下代码段。

BrowserWindow newWindow = new BrowserWindow();

System.Windows.Threading.Dispatcher.Run();

newWindow.Show();

D. 在第 04 行处插入以下代码段。

newWindowThread.SetApartmentState(ApartmentState.STA);

newWindowThread.IsBackground = true;

在第 09 行处插入以下代码段。

BrowserWindow newWindow = new BrowserWindow();

newWindow.Show();

System.Windows.Threading.Dispatcher.Run();

Answer: D

4. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。该应用程序使用若干个异步操作来计算显示给用户的数据。名为 tommorowsWeather 的操作执行将由其他操作使用的计算。

您需要确保 tommorowsWeather 以尽可能高的优先级运行。

您应该使用哪个代码段?

A. tomorrowsWeather.Dispatcher.BeginInvoke(
System.Windows.Threading.DispatcherPriority.Normal,
new OneArgDelegate(UpdateUserInterface),
weather);

B. tomorrowsWeather.Dispatcher.BeginInvoke(
System.Windows.Threading.DispatcherPriority.DataBind,
new OneArgDelegate(UpdateUserInterface),
weather);

C. tomorrowsWeather.Dispatcher.BeginInvoke(
System.Windows.Threading.DispatcherPriority.Send,
new OneArgDelegate(UpdateUserInterface),
weather);

D. tomorrowsWeather.Dispatcher.BeginInvoke(
System.Windows.Threading.DispatcherPriority.Render,
new OneArgDelegate(UpdateUserInterface),
weather);

Answer: C

5. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。

您为该应用程序创建一个窗口。

您需要确保满足以下要求：

È 使用 ListBox 控件以双列格式显示字符串数组。

È ListBox 控件中数据的流动方向为从左向右、自上向下。

您应该怎么办？

A. 使用按如下方式定义的 ListBox 控件。

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

```
  </ListBox.ItemsPanel>
</ListBox>
```

使用下面的 C# 代码将字符串数组与 ListBox 控件关联。

```
myList.ItemsSource = arrayOfString;
```

B. 使用按如下方式定义的 ListBox 控件。

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

使用下面的 C# 代码将字符串数组与 ListBox 控件关联。

```
myList.ItemsSource = arrayOfString;
```

C. 使用按如下方式定义的 ListBox 控件。

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

使用下面的 C# 代码将字符串数组与 ListBox 控件关联。

```
myListView.ItemsSource = arrayOfString;
```

D. 使用按如下方式定义的 ListBox 控件。

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

```
            <ColumnDefinition />
            <ColumnDefinition />
          </Grid.ColumnDefinitions>
        </Grid>
      </ItemsPanelTemplate>
  </ListBox.ItemsPanel>
</ListBox>
```

使用下面的 C# 代码将字符串数组与 ListBox 控件关联。

```
myList.ItemsSource = arrayOfString;
```

Answer: A

6. 您使用 Windows Presentation Foundation 和 Microsoft .NET Framework 3.5 创建一个窗体。该窗体中包含一个状态栏。

您打算向该状态栏添加一个 ProgressBar 控件。

您需要确保 ProgressBar 控件显示您无法预测完成时间的任务的进度。

您应该使用哪个代码段？

A. progbar.IsIndeterminate = true;

B. progbar.IsIndeterminate = false;

C. progbar.HasAnimatedProperties = true;

D. progbar.HasAnimatedProperties = false;

Answer: A

7. 您正在将 Windows 窗体应用程序转换为 Windows Presentation Foundation (WPF) 应用程序。您使用 Microsoft .NET Framework 3.5 创建该 WPF 应用程序。

该 WPF 应用程序将重用此 Windows 窗体应用程序的 30 个窗体。

该 WPF 应用程序中包含下面的类定义。

```
public class OwnerWindow :
  System.Windows.Forms.IWin32Window
{
   private IntPtr handle;
```

```
  public IntPtr Handle
  {
    get { return handle; }
    set { handle=value; }
  }
}
```

您在该 WPF 应用程序中编写以下代码段。(所包括的行号仅供参考。)

```
01 public DialogResult LaunchWindowsFormsDialog(
02   Form dialog, Window wpfParent)
03 {
04    WindowInteropHelper helper=new
05     WindowInteropHelper(wpfParent);
06    OwnerWindow owner=new OwnerWindow();
07
08 }
```

您需要确保该应用程序能够以模式对话框形式启动可重用的窗体。

您应该在第 07 行处插入哪个代码段?

A.
```
owner.Handle = helper.Owner;
return dialog.ShowDialog(owner);
```

B.
```
owner.Handle = helper.Handle;
return dialog.ShowDialog(owner);
```

C.
```
owner.Handle = helper.Owner;
bool? result = wpfParent.ShowDialog();
if (result.HasValue)
 return result.Value   System.Windows.Forms.DialogResult.OK :
 System.Windows.Forms.DialogResult.Cancel;
else
 return System.Windows.Forms.DialogResult.Cancel;
```

D.
```
owner.Handle = helper.Handle;
bool? result = wpfParent.ShowDialog();
```

```
if (result.HasValue)
  return result.Value    System.Windows.Forms.DialogResult.OK :
  System.Windows.Forms.DialogResult.Cancel;
else
  return System.Windows.Forms.DialogResult.Cancel;
```

Answer: B

8. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation (WPF) 应用程序。

该 WPF 应用程序有一个名为 rootGrid 的 Grid 控件。

您编写以下 XAML 代码片段。

```
<Window x:Class="MCP.HostingWinFormsControls"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/
presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="HostingWinFormsControls"
Loaded="Window_Loaded">
   <Grid x:Name="rootGrid">
   </Grid>
</Window>
```

您需要确保每次打开 WPF 窗口时，都会向 rootGrid 中添加一个名为 MyCustomFormsControl 的 Windows 窗体控件。

您应该使用哪个代码段？

A.
```
private void Window_Loaded(object sender, RoutedEventArgs e)
{
   WindowsFormsHost host = new WindowsFormsHost();
   MyCustomFormsControl formsControl = new MyCustomFormsControl();
   host.Child = formsControl;
   rootGrid.Children.Add(host);
}
```

B. private void Window_Loaded(object sender, RoutedEventArgs e)

{

ElementHost host = new ElementHost();

MyCustomFormsControl formsControl=new MyCustomFormsControl();

host.Child=formsControl;

rootGrid.Children.Add(host);

}

C. private void Window_Loaded(object sender, RoutedEventArgs e)

{

MyCustomFormsControl formsControl=new MyCustomFormsControl();

formsControl.CreateControl();

HwndSource source = HwndSource.FromHwnd(formsControl.Handle);

UIElement formsElement = source.RootVisual as UIElement;

rootGrid.Children.Add(formsElement);

}

D. private void Window_Loaded(object sender, RoutedEventArgs e)

{

MyCustomFormsControl formsControl=new MyCustomFormsControl();

formsControl.CreateControl();

HwndTarget target = new HwndTarget(formsControl.Handle);

UIElement formsElement = target.RootVisual as UIElement;

rootGrid.Children.Add(formsElement);

}

Answer: A

9. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。

您在此程序中包含了功能以解决窗口行为问题。

您需要在窗口中鼠标单击的位置显示 UI 元素列表。您还需要确保此元素列表显示在消息框中。

您应该将下面哪个代码段包括在代码隐藏文件中？

A. string controlsToDisplay = string.Empty;

```
private void Window_MouseDown(object sender, MouseButtonEventArgs e) {
   controlsToDisplay = ((UIElement)sender).ToString();
   MessageBox.Show(controlsToDisplay);
}
```

B.

```
string controlsToDisplay = string.Empty;
private void Window_MouseDown(object sender, MouseButtonEventArgs e) {
   for (int i = 0; i < this.VisualChildrenCount; i++) {
      controlsToDisplay + = this.GetVisualChild(i).ToString() + "\r\n";
   }
   MessageBox.Show(controlsToDisplay);
}
```

C.

```
string controlsToDisplay = string.Empty;
private void Window_MouseDown (object sender, MouseButtonEventArgs e)
{
   Visual myVisual;
   for (int i = 0; i < VisualTreeHelper.GetChildrenCount(sender as
     Visual); i++) {
      myVisual = (Visual)VisualTreeHelper.GetChild(sender as Visual, i);
      controlsToDisplay += myVisual.GetType().ToString() + "\r\n";
   }
   MessageBox.Show(controlsToDisplay);
}
```

D.

```
string controlsToDisplay = string.Empty;
private void Window_MouseDown(object sender, MouseButtonEventArgs e) {
   Point pt = e.GetPosition(this);
   VisualTreeHelper.HitTest(this, null, new
     HitTestResultCallback(HitTestCallback), new
     PointHitTestParameters(pt));
   MessageBox.Show(controlsToDisplay);
}
```

```
private HitTestResultBehavior HitTestCallback(HitTestResult result) {
   controlsToDisplay += result.VisualHit.GetType().ToString() + "\r\n";
   return HitTestResultBehavior.Continue;
}
```

Answer: D

10. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。

您在此程序中包含了功能以解决窗口行为问题。

您需要在窗口中鼠标单击的位置显示 UI 元素列表。您还需要确保此元素列表显示在消息框中。

您应该将下面哪个代码段包括在代码隐藏文件中？

A.
```
Dim controlsToDisplay As String = String.Empty
Private Sub Window_MouseDown(ByVal sender As Object, _
 ByVal e As MouseButtonEventArgs)
   controlsToDisplay = CType(sender, UIElement).ToString()
   MessageBox.Show(controlsToDisplay)
End Sub
```

B.
```
Dim controlsToDisplay As String = String.Empty
Private Sub Window_MouseDown(ByVal sender As Object, _
 ByVal e As MouseButtonEventArgs)
   For i = 0 To VisualChildrenCount - 1
      controlsToDisplay += GetVisualChild(i).ToString() + "\r\n"
   Next
   MessageBox.Show(controlsToDisplay)
End Sub
```

C.
```
Dim controlsToDisplay As String = String.Empty
Private Sub Window_MouseDown(ByVal sender As Object, _
ByVal e As MouseButtonEventArgs)
   Dim myVisual As Visual()
   For i = 0 To VisualTreeHelper.GetChildrenCount(CType(sender, _
    Visual)) - 1
```

```
      myVisual(i) = CType(VisualTreeHelper.GetChild(CType(sender, _
        Visual), i), Visual)
      controlsToDisplay += myVisual.GetType().ToString() + "\r\n"
    Next
    MessageBox.Show(controlsToDisplay)
End Sub
```

D.

```
Dim controlsToDisplay As String = String.Empty
Private Sub Window_MouseDown(ByVal sender As Object, _
  ByVal e As MouseButtonEventArgs)
    Dim pt As Point = e.GetPosition(Me)
    VisualTreeHelper.HitTest(Me, Nothing, _
      New HitTestResultCallback(AddressOf HitTestCallback), _
      New PointHitTestParameters(pt))
    MessageBox.Show(controlsToDisplay)
End Sub
Private Function HitTestCallback(ByVal result As HitTestResult) As _
  HitTestResultBehavior
    controlsToDisplay += result.VisualHit.GetType().ToString() + "\r\n"
    Return HitTestResultBehavior.Continue
End Function
```

Answer: D

11. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。您编写以下代码段（所包括的行号仅供参考）。

```
01 Dim content As Object
02 Dim fileName As String = "theFile"
03 Using xamlFile As New FileStream(fileName & ".xaml", _
04 FileMode.Open, FileAccess.Read)
06 content = TryCast(XamlReader.Load(xamlFile), Object)
07 End Using
```

```
08 Using container As Package = Package.Open(fileName & ".xps", _
09 FileMode.Create)
10
11 End Using
```

您需要确保满足以下要求：

È 此应用程序将现有流文档转换为 XPS 文档。

È XPS 文档使用流文档格式生成。

È XPS 文档的大小应为可能的最小值。

您应该在第 10 行插入下面哪个代码段？

A.
```
Using xpsDoc As New XpsDocument(container, _
  CompressionOption.SuperFast)
    Dim rsm As XpsSerializationManager = New _
     System.Windows.Xps.XpsSerializationManager(New _
     XpsPackagingPolicy(xpsDoc), False)
    rsm.SaveAsXaml(paginator)
 End Using
```

B.
```
Using xpsDoc As New XpsDocument(container, _
  CompressionOption.SuperFast)
    Dim rsm As New XpsSerializationManager(New _
     XpsPackagingPolicy(xpsDoc), False)
    rsm.Commit()
 End Using
```

C.
```
Using xpsDoc As New XpsDocument(container, _
  CompressionOption.Maximum)
    Dim rsm As New XpsSerializationManager(New _
     XpsPackagingPolicy(xpsDoc), False)
    Dim paginator As DocumentPaginator = (CType(content, _
     IDocumentPaginatorSource)).DocumentPaginator
    rsm.SaveAsXaml(paginator)
 End Using
```

D. Using xpsDoc As New XpsDocument(container, _

CompressionOption.SuperFast)

Dim rsm As New XpsSerializationManager(New _

XpsPackagingPolicy(xpsDoc), False)

Dim paginator As DocumentPaginator = (CType(content, _

IDocumentPaginatorSource)).DocumentPaginator

rsm.SaveAsXaml(paginator)

End Using

Answer: C

12. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。您编写以下代码段（所包括的行号仅供参考）。

01 Dim content As Object

02 Dim fileName As String = "theFile"

03 Using xamlFile As New FileStream(fileName & ".xaml", _

04 FileMode.Open, FileAccess.Read)

06 content = TryCast(XamlReader.Load(xamlFile), Object)

07 End Using

08 Using container As Package = Package.Open(fileName & ".xps", _

09 FileMode.Create)

10

11 End Using

您需要确保满足以下要求：

È 此应用程序将现有流文档转换为 XPS 文档。

È XPS 文档使用流文档格式生成。

È XPS 文档的大小为可能的最小值。

您应该在第 10 行插入下面哪个代码段？

A. Using xpsDoc As New XpsDocument(container, _

CompressionOption.SuperFast)

Dim rsm As XpsSerializationManager = New _

```
    System.Windows.Xps.XpsSerializationManager(New _
    XpsPackagingPolicy(xpsDoc), False)
  rsm.SaveAsXaml(paginator)
End Using
B. Using xpsDoc As New XpsDocument(container, _
 CompressionOption.SuperFast)
  Dim rsm As New XpsSerializationManager(New _
  XpsPackagingPolicy(xpsDoc), False)
  rsm.Commit()
End Using
C. Using xpsDoc As New XpsDocument(container, _
 CompressionOption.Maximum)
  Dim rsm As New XpsSerializationManager(New _
    XpsPackagingPolicy(xpsDoc), False)
  Dim paginator As DocumentPaginator = (CType(content, _
  IDocumentPaginatorSource)).DocumentPaginator
  rsm.SaveAsXaml(paginator)
End Using
D. Using xpsDoc As New XpsDocument(container, _
 CompressionOption.SuperFast)
  Dim rsm As New XpsSerializationManager(New _
    XpsPackagingPolicy(xpsDoc), False)
  Dim paginator As DocumentPaginator = (CType(content, _
  IDocumentPaginatorSource)).DocumentPaginator
  rsm.SaveAsXaml(paginator)
End Using
```

Answer: C

13. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。此应用程序使用 FlowDocumentPageViewer 类的实例来显示文档。该实例的名称为 fdpv。用户可以突出

显示文档内容并为其添加注释。

您需要确保为文档作的注释得以保存，且这些注释在文档再次显示时呈现出来。

您应该使用下面哪个代码段？

A.
```
protected void OnTextInput(object sender, RoutedEventArgs e) {
   AnnotationService service = AnnotationService.GetService(fdpv);
   if (service == null) {
      AnnotationStream = new FileStream("annotations.xml",
        FileMode.Open, FileAccess.ReadWrite);
      service = new AnnotationService(fdpv);
      AnnotationStore store = new XmlStreamStore(AnnotationStream);
      service.Enable(store);
   }
}
private void OnClosing(object sender,
  System.ComponentModel.CancelEventArgs e) {
   AnnotationService service = AnnotationService.GetService(fdpv);
   if (service != null && service.IsEnabled) {
      service.Store.Flush();
      service.Disable();
      AnnotationStream.Close();
   }
}
```

B.
```
protected void OnLoaded(object sender, RoutedEventArgs e) {
   AnnotationService service = AnnotationService.GetService(fdpv);
   if (service == null) {
      AnnotationStream = new FileStream("annotations.xml",
        FileMode.Open, FileAccess.ReadWrite);
      service = new AnnotationService(fdpv);
   }
}
```

```
private void OnClosing(object sender,
  System.ComponentModel.CancelEventArgs e) {
    AnnotationService service = AnnotationService.GetService(fdpv);
    if (service != null && service.IsEnabled) {
        service.Store.Flush();
        service.Disable();
        AnnotationStream.Close();
    }
}
```

C.

```
protected void OnLoaded(object sender, RoutedEventArgs e) {
    AnnotationService service = AnnotationService.GetService(fdpv);
    if (service == null) {
        AnnotationStream = new FileStream("annotations.xml",
          FileMode.Open, FileAccess.ReadWrite);
        service = new AnnotationService(fdpv);
        AnnotationStore store = new XmlStreamStore(AnnotationStream);
        service.Enable(store);
    }
}
private void OnClosing(object sender,
  System.ComponentModel.CancelEventArgs e) {
    AnnotationService service = AnnotationService.GetService(fdpv);
    if (service != null && service.IsEnabled) {
        service.Store.Flush();
        service.Disable();
        AnnotationStream.Close();
    }
}
```

D.

```
protected void OnLoaded(object sender, RoutedEventArgs e) {
    AnnotationService service = AnnotationService.GetService(fdpv);
```

```
    if (service == null) {
        AnnotationStream = new FileStream("annotations.xml",
        FileMode.Open, FileAccess.ReadWrite);
        service = new AnnotationService(fdpv);
        AnnotationStore store = new XmlStreamStore(AnnotationStream);
        service.Enable(store);
    }
}
private void OnClosing(object sender,
  System.ComponentModel.CancelEventArgs e) {
    AnnotationService service = AnnotationService.GetService(fdpv);
    if (service != null && service.IsEnabled) {
        service.Disable();
        AnnotationStream.Close();
    }
}
```

Answer: C

14. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。此应用程序将使用 FlowDocumentPageViewer 类的实例来显示文档。该实例的名称为 fdpv。用户可以突出显示文档内容并为其添加注释。

您需要确保为文档作的注释得以保存，且这些注释在文档再次显示时呈现出来。

您应该使用下面哪个代码段？

A.
```
Protected Sub OnTextInput(ByVal sender As Object, _
ByVal e As RoutedEventArgs)
Dim service As AnnotationService = _
AnnotationService.GetService(fdpv)
If service Is Nothing Then
AnnotationStream = New FileStream("annotations.xml", _
FileMode.Open, FileAccess.ReadWrite)
```

```
service = New AnnotationService(fdpv)
Dim store As AnnotationStore = _
New XmlStreamStore(AnnotationStream)
service.Enable(store)
End If
End Sub
Private Sub OnClosing(ByVal sender As Object, _
ByVal e As System.ComponentModel.CancelEventArgs)
Dim service As AnnotationService = _
AnnotationService.GetService(fdpv)
If Not service Is Nothing AndAlso service.IsEnabled Then
service.Store.Flush()
service.Disable()
AnnotationStream.Close()
End If
End Sub
```

B.

```
Protected Sub OnLoaded(ByVal sender As Object, _
ByVal e As RoutedEventArgs)
Dim service As AnnotationService = _
AnnotationService.GetService(fdpv)
If service Is Nothing Then
AnnotationStream = New FileStream("annotations.xml", _
FileMode.Open, FileAccess.ReadWrite)
service = New AnnotationService(fdpv)
End If
End Sub
Private Sub OnClosing(ByVal sender As Object, _
ByVal e As System.ComponentModel.CancelEventArgs)
Dim service As AnnotationService = _
AnnotationService.GetService(fdpv)
```

```
If Not service Is Nothing AndAlso service.IsEnabled Then
service.Store.Flush()
service.Disable()
AnnotationStream.Close()
End If
End Sub
```

C.

```
Protected Sub OnLoaded(ByVal sender As Object, _
ByVal e As RoutedEventArgs)
Dim service As AnnotationService = _
AnnotationService.GetService(fdpv)
If service Is Nothing Then
AnnotationStream = New FileStream("annotations.xml", _
FileMode.Open, FileAccess.ReadWrite)
service = New AnnotationService(fdpv)
Dim store As AnnotationStore = New _
XmlStreamStore(AnnotationStream)
service.Enable(store)
End If
End Sub
Private Sub OnClosing(ByVal sender As Object, _
ByVal e As System.ComponentModel.CancelEventArgs)
Dim service As AnnotationService = _
AnnotationService.GetService(fdpv)
If Not service Is Nothing AndAlso service.IsEnabled Then
service.Store.Flush()
service.Disable()
AnnotationStream.Close()
End If
End Sub
```

D.

```
Protected Sub OnLoaded(ByVal sender As Object, _
```

ByVal e As RoutedEventArgs)

Dim service As AnnotationService = _

AnnotationService.GetService(fdpv)

If service Is Nothing Then

AnnotationStream = New FileStream("annotations.xml", _

FileMode.Open, FileAccess.ReadWrite)

service = New AnnotationService(fdpv)

Dim store As AnnotationStore = New _

XmlStreamStore(AnnotationStream)

service.Enable(store)

End If

End Sub

Private Sub OnClosing(ByVal sender As Object, _

ByVal e As System.ComponentModel.CancelEventArgs)

Dim service As AnnotationService = _

AnnotationService.GetService(fdpv)

If Not service Is Nothing AndAlso service.IsEnabled Then

service.Disable()

AnnotationStream.Close()

End If

End Sub

Answer: C

15. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。您打算使用此应用程序预览视频文件。

您编写以下 XAML 代码片段。

01 <Window

01 x:Class="myClass" xmlns=

01 "http://schemas.microsoft.com/winfx/2006/xaml/presentation"

01 xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

```
01 Title="myWindow" Height="300" Width="300">
02     <StackPanel Background="Black">
03
04         <StackPanel HorizontalAlignment="Center"
04         Orientation="Horizontal">
05             <Button Name="btnPlay" Margin="10" Content="Play" />
06         </StackPanel>
07
08     </StackPanel>
09 </Window>
```

您需要确保此应用程序仅播放您要预览的视频的前 10 秒。

您应该执行哪两个操作？（每个正确的答案提供部分解决方案。请选择两个答案。）

A. 在第 03 行插入以下 XAML 片段。

```
<MediaElement Name="myMediaElement" Stretch="Fill" />
```

B. 在第 03 行插入以下 XAML 片段。

```
<MediaElement Name="myMediaElement"
Source="MediaFileSelected.wmv" Stretch="Fill" />
```

C. 在代码隐藏文件中创建以下方法。

```
public void PlayMedia(object sender, RoutedEventArgs args) {
   myMediaElement.Play();
}
```

D. 在第 07 行插入以下 XAML 片段。

```
<StackPanel.Triggers>
   <EventTrigger RoutedEvent="Button.Click" SourceName="btnPlay">
      <EventTrigger.Actions>
         <BeginStoryboard Name= "myBegin">
            <Storyboard SlipBehavior="Slip">
               <MediaTimeline Source="MediaFileSelected.wmv"
               Storyboard.TargetName="myMediaElement"
               BeginTime="0:0:0" Duration="0:0:10" />
```

```
            </Storyboard>
          </BeginStoryboard>
        </EventTrigger.Actions>
    </EventTrigger>
</StackPanel.Triggers>
```

E. 在第 07 行插入以下 XAML 片段。

```
<StackPanel.Triggers>
    <EventTrigger RoutedEvent="Button.Click" SourceName="btnPlay">
        <EventTrigger.Actions>
            <BeginStoryboard Name= "myBegin">
                <Storyboard SlipBehavior="Slip">
                    <MediaTimeline
                    Storyboard.TargetName="myMediaElement"
                    BeginTime="0:0:0" Duration="0:0:10" />
                </Storyboard>
            </BeginStoryboard>
        </EventTrigger.Actions>
    </EventTrigger>
</StackPanel.Triggers>
```

Answer: AD

16. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。您打算使用此应用程序预览视频文件。

您编写以下 XAML 代码片段。

```
01 <Window
01 x:Class="myClass" xmlns=
01 "http://schemas.microsoft.com/winfx/2006/xaml/presentation"
01 xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
01 Title="myWindow" Height="300" Width="300">
02     <StackPanel Background="Black">
```

```
03
04        <StackPanel HorizontalAlignment="Center"
04        Orientation="Horizontal">
05           <Button Name="btnPlay" Margin="10" Content="Play" />
06        </StackPanel>
07
08     </StackPanel>
09 </Window>
```

您需要确保此应用程序仅播放您要预览的视频的前 10 秒。

您应该执行哪两个操作？（每个正确的答案提供部分解决方案。请选择两个答案。）

A. 在第 03 行插入以下 XAML 片段。

```
<MediaElement Name="myMediaElement" Stretch="Fill" />
```

B. 在第 03 行插入以下 XAML 片段。

```
<MediaElement Name="myMediaElement"
Source="MediaFileSelected.wmv" Stretch="Fill" />
```

C. 在代码隐藏文件中创建以下方法。

```
Public Sub PlayMedia(ByVal sender As Object, _
ByVal args As RoutedEventArgs)
        myMediaElement.Play()
End Sub
```

D. 在第 07 行插入以下 XAML 片段。

```
<StackPanel.Triggers>
   <EventTrigger RoutedEvent="Button.Click" SourceName="btnPlay">
      <EventTrigger.Actions>
         <BeginStoryboard Name= "myBegin">
            <Storyboard SlipBehavior="Slip">
               <MediaTimeline Source="MediaFileSelected.wmv"
               Storyboard.TargetName="myMediaElement"
               BeginTime="0:0:0" Duration="0:0:10" />
            </Storyboard>
```

```
        </BeginStoryboard>
      </EventTrigger.Actions>
    </EventTrigger>
</StackPanel.Triggers>
```

E. 在第 07 行插入以下 XAML 片段。

```
<StackPanel.Triggers>
    <EventTrigger RoutedEvent="Button.Click" SourceName="btnPlay">
      <EventTrigger.Actions>
        <BeginStoryboard Name= "myBegin">
          <Storyboard SlipBehavior="Slip">
            <MediaTimeline
            Storyboard.TargetName="myMediaElement"
            BeginTime="0:0:0" Duration="0:0:10" />
          </Storyboard>
        </BeginStoryboard>
      </EventTrigger.Actions>
    </EventTrigger>
</StackPanel.Triggers>
```

Answer: AD

17. 您正在创建一个 Windows Presentation Foundation 应用程序。

您为此应用程序创建一个窗口。此应用程序包含一个名为 AudioFileToPlay.wav 的音频文件。

您需要确保每次单击该窗口的工作区时都会播放该音频文件。

您应该怎么办?

A. 将以下 XAML 代码行添加到该窗口中。

```
<MediaElement Source="AudioFileToPlay.wav" />
```

B. 将以下代码段添加到代码隐藏文件中的 window constructor 方法中。

```
SoundPlayer player = new SoundPlayer();
player.SoundLocation = "AudioFileToPlay.wav";
player.Play();
```

C. 将以下代码段添加到代码隐藏文件中的 window MouseDown 方法中。

```
MediaPlayer player = new MediaPlayer();
player.SetValue(MediaElement.SourceProperty,new Uri("AudioFileToPlay.wav", UriKind.Relative));
player.Play();
```

D. 将以下 XAML 代码片段添加到窗口中。

```
<Window.Triggers>
  <EventTrigger RoutedEvent="Window.MouseDown">
    <EventTrigger.Actions>
      <SoundPlayerAction Source="AudioFileToPlay.wav"/>
    </EventTrigger.Actions>
  </EventTrigger>
</Window.Triggers>
```

Answer: D

18. 您正在创建一个 Windows Presentation Foundation 应用程序。

您为此应用程序创建一个窗口。此应用程序包含一个名为 AudioFileToPlay.wav 的音频文件。

您需要确保满足以下要求：

È 每次单击该窗口的工作区时都会播放该音频文件。

È 该窗口在音频文件播放时提供最佳性能。

您应该怎么办？

A. 将以下 XAML 代码行添加到该窗口中。

```
<MediaElement Source="AudioFileToPlay.wav" />
```

B. 将以下代码段添加到代码隐藏文件中的 window constructor 方法中。

```
Dim player As New SoundPlayer()
player.SoundLocation = "AudioFileToPlay.wav"
player.Play()
```

C. 将以下代码段添加到代码隐藏文件中的 window MouseDown 方法中。

```
Dim player As New MediaElement()
player.Source = New Uri("AudioFileToPlay.wav", UriKind.Relative)
player.LoadedBehavior = MediaState.Manual
```

player.Play()

D. 将以下 XAML 代码片段添加到窗口中。

```
<Window.Triggers>
  <EventTrigger RoutedEvent="Window.MouseDown">
    <EventTrigger.Actions>
      <SoundPlayerAction Source="AudioFileToPlay.wav"/>
    </EventTrigger.Actions>
  </EventTrigger>
</Window.Triggers>
```

Answer: D

19. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。

您的项目包含一个名为 Data 的文件夹。

您在此 Data 文件夹中添加一个名为 song.mp3 的 MP3 文件。您将此 MP3 文件的 Build Action 属性设置为 Resource。

您需要从该应用程序访问此 MP3 文件。

您应该使用下面哪个代码段？

A.
```
Uri uri = new Uri("/Data/song.mp3", UriKind.Relative);
StreamResourceInfo sri=Application.GetContentStream(uri);
Stream stream=sri.Stream;
```

B.
```
Uri uri = new Uri("/Data/song.mp3", UriKind.Relative);
StreamResourceInfo sri=Application.LoadComponent(uri);
Stream stream=sri.Stream;
```

C.
```
Uri uri = new Uri("/Data/song.mp3", UriKind.Relative);
StreamResourceInfo sri=Application.GetRemoteStream(uri);
Stream stream=sri.Stream;
```

D.
```
Uri uri = new Uri("/Data/song.mp3", UriKind.Relative);
StreamResourceInfo sri=Application.GetResourceStream(uri);
Stream stream=sri.Stream;
```

Answer: D

20. 您正在使用 Microsoft .NET Framework 3.5 创建一个 Windows Presentation Foundation 应用程序。

您的项目包含一个名为 Data 的文件夹。

您在此 Data 文件夹中添加一个名为 song.mp3 的 .MP3 文件。您将此应用程序的 Build Action 属性设置为 Resource。

您需要通过其中一个 Application 类访问此 .MP3 文件。

您应该使用哪个代码段？

A. Dim uri As New Uri("/Data/song.mp3", UriKind.Relative)

Dim sri As StreamResourceInfo = Application.GetContentStream(uri)

Dim stream As Stream = sri.Stream

B. Dim uri As New Uri("/Data/song.mp3", UriKind.Relative)

Dim sri As StreamResourceInfo = Application.LoadComponent(uri)

Dim stream As Stream = sri.Stream

C. Dim uri As New Uri("/Data/song.mp3", UriKind.Relative)

Dim sri As StreamResourceInfo = Application.GetRemoteStream(uri)

Dim stream As Stream = sri.Stream

D. Dim uri As New Uri("/Data/song.mp3", UriKind.Relative)

Dim sri As StreamResourceInfo = Application.GetResourceStream(uri)

Dim stream As Stream = sri.Stream

Answer: D