



**Vendor: Microsoft**

**Exam Code: 70-484**

**Exam Name: Microsoft Essentials of Developing Windows**

**Store Apps using C#**

**Version: Demo**

## Case Study: 1 - Scenario 1

### Background

You are developing a Windows Store app by using C# and XAML. The app will allow users to share and rate photos. The app will also provide information to users about photo competitions.

### Application Structure

The app stores data by using a class that is derived from the DataStoreBase class.

The app coordinates content between users by making calls to a centralized RESTful web service.

The app has a reminder system that displays toast notifications when a photo competition is almost over. The app gets the competition schedule data from the web service.

The app displays a list of images that are available for viewing in a data-bound list box. The image file list stores paths to the image files. The app downloads new images from the web service on a regular basis.

Relevant portions of the app files are shown. (Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

### Business Requirements

The app must allow users to do the following:

- Run the app on a variety of devices, including devices that have limited bandwidth connections.
- Share and synchronize high resolution photographs that are greater than 1 MB in size.
- Rate each photo on a scale from 1 through 5.

### Technical Requirements

The app must meet the following technical requirements:

- Retain state for each user and each device.
- Restore previously saved state each time the app is launched.
- Preserve user state and photo edits when switching between this app and other apps.
- When the app resumes after a period of suspension, refresh the user interface, tile images, and data with current information from the web service.
- Update the image list box as new images are added to the image file list.
- Convert the image paths into images when binding the image file list to the list box.

The app must store cached images on the device only, and must display images or notifications on the app tile to meet the following requirements:

- Regularly update the app tile with random images from the user's collection displayed one at a time.
- When a photo is displayed on the tile, one of the following badges must be displayed:
  - If the photo has a user rating, the tile must display the average user rating as a badge.
  - If the photo does not have a rating, the tile must display the Unavailable glyph as a badge.
- Update the app tile in real time when the app receives a notification.
- Display only the most recent notification on the app tile.

The app must display toast notifications to signal the end of a photo competition.

The toast notifications must meet the following requirements:

- Display toast notifications based on the schedule that is received from the web service.
- Display toast notifications for as long as possible.
- Display toast notifications regardless of whether the app is running.
- When a user clicks a toast notification that indicates the end of the photo competition, the app must display the details of the photo competition that triggered the toast notification.

#### App.xaml.cs

```

AX01 using System;
AX02 using Windows.ApplicationModel.Activation;
AX03 using Windows.UI.Xaml;
AX04 using Windows.UI.Xaml.Controls;
AX05 namespace Application1
AX06 {
AX07     sealed partial class App : Application
AX08     {
AX09         private DispatcherTimer tileUpdateTimer = new DispatcherTimer();
AX10         private DispatcherTimer badgeUpdateTimer = new DispatcherTimer();
AX11         public App()
AX12         {
AX13             this.InitializeComponent();
AX14             tileUpdateTimer.Tick += TileUpdateTimer_Tick;
AX15             tileUpdateTimer.Interval = new TimeSpan(0, 0, 10);
AX16             tileUpdateTimer.Start();
AX17             badgeUpdateTimer.Tick += BadgeUpdateTimer_Tick;
AX18             badgeUpdateTimer.Interval = new TimeSpan(0, 1, 0);
AX19             badgeUpdateTimer.Start();
AX20         }
AX21         private void SendNotification(XmlDocument currentTemplate)
AX22         {
AX23             var tileUpdater = TileUpdateManager.CreateTileUpdaterForApplication();
AX24             tileUpdater.Update(currentTemplate);
AX25         }
AX26         void TileUpdateTimer_Tick(object sender, object e)
AX27         {
AX28             SendNotification(currentTemplate);
AX29         }
AX30         void BadgeUpdateTimer_Tick(object sender, object e)
AX31         {
AX32             ...
AX33         }
AX34         protected override void OnLaunched(LaunchActivatedEventArgs args)
AX35         {
AX36             var rootFrame = new Frame();
AX37             rootFrame.Navigate(typeof(MainPage));
AX38             Window.Current.Content = rootFrame;
AX39             Window.Current.Activate();
AX40         }
AX41     }
AX42 }

```

### DataStoreBase.cs

```
DB01 using System;
DB02 namespace Application1
DB03 {
DB04     public abstract class DataStoreBase
DB05     {
DB06         public abstract bool SaveLocalSetting(string key, string value);
DB07         public abstract bool SaveRoamingSetting(string key, string value);
DB08         public abstract bool SaveDataToWebService(string key, string jsonString);
DB09         public abstract bool SaveDataToLocalStorage(string key, string jsonString);
DB10         public abstract bool SaveDataToRoamingStorage(string key, string jsonString);
DB11         public abstract bool SaveDataToAzureStorage(string key, string jsonString);
DB12         public abstract string GetLocalSetting(string key);
DB13         public abstract string GetRoamingSetting(string key);
DB14         public abstract string GetDataFromWebService(string key);
DB15         public abstract string GetDataFromLocalStorage(string key);
DB16         public abstract string GetDataFromRoamingStorage(string key);
DB17         public abstract string GetDataFromAzureStorage(string key);
DB18     }
DB19 }
```

### QUESTION 1

You need to ensure that launching the app displays the required information. From which ApplicationExecutionState enumeration should you configure the user interface state?

- A. ClosedByUser
- B. Suspended
- C. NotRunning
- D. Running
- E. Terminated

**Correct Answer:** A

#### Explanation:

\* ClosedByUser

The user closes the app through the close gesture or Alt+F4 and takes longer than 10 seconds to activate the app again.

\*From scenario: The app must meet the following technical requirements:

Retain state for each user and each device.

Restore previously saved state each time the app is launched.

### QUESTION 2

You need to choose the appropriate data binding strategy for the image list box. Which method should you use?

- A. System.Drawing.ImageConverter.ConvertToString()
- B. IValueConverter.ConvertBack()
- C. IValueConverter.Convert()
- D. System.Drawing.ImageConverter.ConvertFromStrin()

**Correct Answer:** C

#### Explanation:

IValueConverter.Convert

The data binding engine calls this method when it propagates a value from the binding source to the binding target.

### QUESTION 3

You need to configure toast notifications for the photo competition.  
Which code segment should you use?

- A. 

```
((XmlElement)currentTemplate.CreateElement("notify")).SetAttribute("duration", "5000");
```
- B. 

```
((XmlElement)currentTemplate.GetElementsByTagName("toast")[0]).SetAttribute("duration", "long");
```
- C. 

```
((XmlElement)currentTemplate.GetElementsByTagName("toast")[0]).SetAttribute("duration", "short");
```
- D. 

```
((XmlElement)currentTemplate.CreateElement("duration")).SetAttribute("value", "long");
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Correct Answer:** B

**Explanation:**

The app must display toast notifications to signal the end of a photo competition. The toast notifications must meet the following requirements:

/ Display toast notifications for as long as possible

### QUESTION 4

You need to ensure that only the correct information is preserved when the user switches to another app. Which actions should you perform? (Each correct answer presents part of the solution. Choose all that apply.)

- A. Save application state by calling the SaveDataToRoamingStorage() method,
- B. Save photographs by calling the SaveDataToLocalStorage() method.
- C. Save photographs by calling the SaveDataToWebService() method.
- D. save application state by calling the SaveDataToLocalStorage() method.

**Correct Answer:** AB

**Explanation:**

A: From scenario: The app must meet the following technical requirements:

Retain state for each user and each device.

B: From scenario: The app must store cached images on the device only

### QUESTION 5

You need to ensure that the app resumes according to the requirements.

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

- A. Retrieve new user content by using the Window.Current.Dispatcher.ProcessEvents() method in the App\_Resuming event handler.
- B. update the user interface by using the Window.Current.Dispatcher.Invoke() method in the

- App\_Resuming event handler.
- C. Override the OnLaunched event handler.
- D. Retrieve new user content by using the Window.Current.Dispatcher.ProcessEvents() method in the OnLaunched event handler when the ActivationKind is Launch.
- E. Update the user interface by using the Window.Current.Dispatcher.Invoke() method in the OnLaunched event handler when the ActivationKind is Launch.
- F. Register the App\_Resuming event handler for theResuming event.

**Correct Answer:** AB

**Explanation:**

From scenario:

The app must meet the following technical requirements:

/ When the app resumes after a period of suspension, refresh the user interface, tile images, and data with current information from the web service.

**QUESTION 6**

You need to choose the appropriate data binding strategy for the image list box.  
Which method should you use?

- A. System.Drawing.ImageConverter.ConvertTo(value, typeof(Image))
- B. IValueConverter.Convert()
- C. System.Drawing.ImageConverter.ConvertFrom(value, typeof(Image), CultureInfo.CurrentUICulture)
- D. IValueConverter.ConvertBack()

**Correct Answer:** B

**Explanation:**

IValueConverter.Convert

The data binding engine calls this method when it propagates a value from the binding source to the binding target.

**QUESTION 7**

A photo competition is ending.

You need to meet the requirements when a user clicks the toast notification.

Which code segment should you use?

- A. 

```
((XmlElement)currentTemplate.GetElementsByTagName("binding")[0])  
.SetAttribute("trigger", competitionID);
```
- B. 

```
currentTemplate.GetElementsByTagName("binding").First()  
.AppendChild(currentTemplate.CreateTextNode(competitionID));
```
- C. 

```
currentTemplate.GetElementsByTagName("toast").First()  
.AppendChild(currentTemplate.CreateTextNode(competitionID));
```
- D. 

```
((XmlElement)currentTemplate.GetElementsByTagName("toast")[0])  
.SetAttribute("launch", competitionID);
```

- A. Option A
- B. Option B
- C. Option C

D. Option D

**Correct Answer:** D

**Explanation:**

From scenario:

When a user clicks a toast notification that indicates the end of the photo competition, the app must display the details of the photo competition that triggered the toast notification.

### QUESTION 8

You need to identify the class to use as the data context for the image list box. Which class should you use?

- A. System.Collections.ObjectModel.CollectionObserver<T>
- B. System.Collections.ObjectModel.ObservableCollection<T>
- C. System.Collections.Generic.IEnumerator<T>
- D. System.Collections.Generic.IEnumerable<T>

**Correct Answer:** B

**Explanation:**

ObservableCollection<T> Class

Represents a dynamic data collection that provides notifications when items get added, removed, or when the whole list is refreshed.

### QUESTION 9

You need to correctly display notifications on the app tile.

Which code segments should you insert at line AX24? (Each correct answer presents part of the solution.

Choose all that apply.)

- A. `tileUpdater.EnableNotificationQueue(false);`
- B. `tileUpdater.EnableNotificationQueue(true);`
- C. `tileUpdater.Update(new TileNotification(currentTemplate)  
{ ExpirationTime = DateTimeOffset.Now.AddMinutes(10) });`
- D. `tileUpdater.AddToSchedule(new ScheduledTileNotification(  
currentTemplate, DateTimeOffset.Now.AddMinutes(10)));`

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Correct Answer:** BD

**Explanation:**

B: TileUpdater.EnableNotificationQueue | enableNotificationQueue method Enables the tile to queue up to five notifications. This enables the notification queue on all tile sizes.

D: From scenario:

Display toast notifications based on the schedule that is received from the web service.

### QUESTION 10

You need to ensure that resuming the app displays the required information.  
From which ApplicationExecutionState enumeration should you configure the user interface state?

- A. Terminated
- B. NotRunning
- C. Suspended
- D. Running
- E. ClosedByUser

**Correct Answer:** E

**Explanation:**

\* ClosedByUser

The user closes the app through the close gesture or Alt+F4 and takes longer than 10 seconds to activate the app again.

\*From scenario: The app must meet the following technical requirements:

Retain state for each user and each device.

Restore previously saved state each time the app is launched.

### QUESTION 11

You need to ensure that only the correct information is preserved when the user switches to another app.  
Which actions should you perform? (Each correct answer presents part of the solution. Choose all that apply.)

- A. Save photographs by calling the SaveDataToAzureStorage() method.
- B. Save application state by calling the SaveDataToLocalStorageQ method.
- C. Save application state by calling the SaveDataToRoamingStorage() method,
- D. Save photographs by calling the SaveDataToWebService() method.
- E. Save application state by calling the SaveDataToAzureStorage() method.
- F. Save photographs by calling the SaveDataToLocalStorage() method.

**Correct Answer:** EF

**Explanation:**

C: The app must meet the following technical requirements:

Retain state for each user and each device.

Restore previously saved state each time the app is launched.

F: The app must store cached images on the device only

### QUESTION 12

You need to access the remote image data according to the requirements.  
Which data storage methods should you use?

- A. SaveDataToSqlAzureStorage() and GetDataFromSqlAzureStorage()
- B. SaveDataToRemoteStorage() and GetDataFromRemoteStorage()
- C. SaveDataToAzureStorage() and GetDataFromAzureStorage()
- D. SaveDataToWebService() and GetDataFromWebService()

**Correct Answer:** C



**QUESTION 13**

You need to update the app tile images.

With which four code segments in sequence should you replace line AX23? (To answer, move the appropriate code segments from the list of code segments to the answer area and arrange them in the correct order.)

*****	
	Answer Area
<pre>var tileUpdater = TileUpdateManager.CreateTileUpda terForApplication(); tileUpdater.Insert (new TileNotification (currentTemplate));</pre>	
<pre>currentTemplate = TileUpdateManager.GetTemplateCon tent (TileTemplateType.TileWideImage) ;</pre>	
<pre>currentTemplate.GetXml();</pre>	
<pre>var tileUpdater = TileUpdateManager.CreateTileUpda terForApplication(); tileUpdater.Update (new TileNotification (currentTemplate));</pre>	
<pre>imageNode.SetAttribute ("src", string.Format("ms- appdata://{0}", GetRandomImage ()));</pre>	
<pre>var imageNode = (XmlElement) currentTemplate.GetElementsByTagName ("image")[0];</pre>	
<pre>imageNode.SetAttribute ("src", string.Format("ms- appx://{0}", GetRandomImage()));</pre>	

**Correct Answer:**

Answer Area	
<pre>var tileUpdater = TileUpdateManager.CreateTileUpda terForApplication(); tileUpdater.Insert (new TileNotification (currentTemplate));</pre>	<pre>currentTemplate.GetXml();</pre>
<pre>currentTemplate = TileUpdateManager.GetTemplateCon tent (TileTemplateType.TileWideImage) ;</pre>	<pre>var tileUpdater = TileUpdateManager.CreateTileUpda terForApplication(); tileUpdater.Update (new TileNotification (currentTemplate));</pre>
	<pre>var imageNode = (XmlElement) currentTemplate.GetElementsByTag Name("image")[0];</pre>
	<pre>imageNode.SetAttribute ("src", string.Format("ms- appdata://{0}", GetRandomImage ()));</pre>
<pre>imageNode.SetAttribute ("src", string.Format("ms- appx://{0}", GetRandomImage()));</pre>	

**Explanation:**

\* From scenario:

/ Regularly update the app tile with random images from the user's collection displayed one at a time.

\* What is the difference between ms-appx:/// and ms-appdata:///

The bird's eye view is that ms-appx is your app package (where the app is installed), and ms-appdata is your application data .

## Case Study: 2 - Scenario 2

### Background

You are developing a Windows Store app named Picture Sharer. The app will allow users to capture, modify, caption, and share pictures.

### Application Structure

The ShareImageButton and GetContactsButton controls use the same foreground color. The foreground color might change in the future.

The following code defines a custom button style named ButtonStyleRed:

```
<StyleTargetType="Button"x:Key="ButtonStyleRed">
  <SetterProperty="Foreground"Value="#FFC34343"/>
  <SetterProperty="BorderBrush"Value="#FFC34343"/>
  ...
</Style>
```

Relevant portions of the app files are shown. (Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

### Business Requirements

The app must meet the following business requirements:

- Allow users to capture and retrieve pictures, modify pictures by adding a shading effect, and add captions to images.
- Support only Landscape and Landscape-flipped orientations.
- Ensure that users can select and modify images from the PictureChooserPage page.
- Ensure that users can change the magnification of the selected image and resize the image by using pinch and stretch gestures. Scaling should be fluid and precisely controlled by the user.

The app must be localized for the French Canadian market.

### Technical Requirements

The app must meet the following technical requirements:

- Scroll bars must not be visible.
- The CaptionTextBlock and CaptionTextBox controls must appear side by side, without overlapping and on the same line. The CaptionTextBox control should appear to the right of the CaptionTextBlock control.
- The ContactPicker object must be filtered to display only email addresses.
- Minimize the code that is required to implement optical zoom functionality.

You must perform the following tasks:

- Handle the Click event of the GetPictureButton control to switch from the current page to the PictureChooserPage page.
- After the user selects an image on the PictureChooserPage page, ensure that the app navigates back to the PictureSharerMainPage page.
- Track the current screen orientation and page size by updating the \_currentViewState, \_currentHeight, and \_currentWidth fields every time the screen orientation or page size changes.

- Create a style named ButtonStyleWhite that inherits all the style settings of the ButtonStyleRed style except the border color; the border color must be white. The ButtonStyleWhite style must automatically update with any changes that are made to the ButtonStyleRed style.
- Create a resource named ButtonForegroundColor to implement the button foreground color so that it can be referenced in XAML by using the following standard syntax: `Foreground="{StaticResource ButtonForegroundColor}"`
- Ensure that the `OnNavigatedTo()` method updates the current picture when a new picture is selected.
- Change the background for the root Grid element to a vertical gradient that transitions from black at the top to maroon at the bottom. Create a resource named GridBackgroundGradientBrush to hold the requested gradient.

While testing the app, you observe the following results:

- An exception is being thrown in the `GetContactsCompleted` event handler when the retrieved email address is assigned to the `RecipientsTextBlock` control. The exception message states: "The application called an interface that was marshalled for a different thread."
- When users navigate away from the `PictureSharerMainPage` page, information that was entered in the `CaptionTextBox` control is lost.

#### PictureSharerMainPage.xaml

```

XA01 <Page
XA02     x:Class="PictureSharer.PictureSharerMainPage"
XA03     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
XA04     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
XA05     xmlns:local="using:PictureSharer"
XA06     xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
XA07     xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
XA08     mc:Ignorable="d">
XA09
XA10     <GridBackground="{StaticResourceApplicationPageBackgroundBrush}">
XA11         <Imagex>Name="SelectedImage"Source="Images/blank.jpg"Width="800"Height="800" />
XA12         <TextBlockx>Name="RecipientsTextBlock"/>
XA13         <StackPanel>
XA14             <TextBlockx>Name="CaptionTextBlock"Text="Caption"/>
XA15             <TextBoxx>Name="CaptionTextBox"/>
XA16         </StackPanel>
XA17         <Buttonx>Name="ShareImageButton"Click="ShareImageButton_Click"
Foreground="#FFC34343">
XA18             Send Image</Button>
XA19         <Buttonx>Name="GetContactsButton"Click="GetContactsButton_Click"
Foreground="#FFC34343">
XA20             Get Contacts</Button>
XA21         <Buttonx>Name="GetPictureButton"Click="GetPictureButton_Click"
Foreground="#FFC34343" >
XA22             Get Picture</Button>
XA23     </Grid>
XA24 </Page>

```

**PictureSharerMainPage.xaml.cs**

```

CS01 publicsealedpartialclassPictureSharerMainPage: Page
CS02 {
CS03     private ApplicationViewState _currentViewState;
CS04     private double _currentHeight, _currentWidth;
CS05     publicPictureSharerMainPage()
CS06     {
CS07         this.InitializeComponent();
CS08     }
CS09 }
CS10 }
CS11 protectedoverridevoidOnNavigatedTo(NavigationEventArgs)
CS12 {
CS13 }
CS14 }
CS15 privatevoidGetContactsButton_Click(objectsender, RoutedEventArgs)
CS16 {
CS17     varpicker = newContactPicker();
CS18 }
CS19     varresults = picker.PickSingleContactAsync();
CS20     results.Completed += GetContactsCompleted;
CS21 }
CS22 privatevoidGetContactsCompleted(IAsyncOperation<ContactInformation> op,
CS23     AsyncStatusstatus)
CS24 {
CS25     varemailList = newList<string>();
CS26     varcontact = op.GetResults();
CS27     if(contact.Emails.Count == 0)
CS28         return;
CS29     foreach(varinfo incontact.Emails)
CS30         emailList.Add(info.Value);
CS31     varemail = string.Join(";", emailList);
CS32     RecipientsTextBlock.Text = email;
CS33 }
CS34 privatevoidShareImageButton_Click(objectsender, RoutedEventArgs)
CS35 {
CS36     SendImageToCloud();
CS37 }
CS38 privatevoidSendImageToCloud()
CS39 {
CS40     ...
CS41 }
CS42 }
CS43 }

```

**PictureChooserPage.xaml**

```

PC01 <Page
PC02     x:Class="PictureSharer.PictureChooserPage"
PC03     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
PC04     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
PC05     xmlns:local="using:PictureSharer"
PC06     xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
PC07     xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
PC08     mc:Ignorable="d">
PC09     <GridBackground="{StaticResourceApplicationPageBackgroundBrush}">
PC10         <Imagex:Name="SelectedImage"Source="Images/image1234.jpg"/>
PC11         <ButtonContent="Back"HorizontalAlignment="Left"Margin="227,25,0,0"
PC12             VerticalAlignment="Top"Width="75" Click="Button_Click_1"/>
PC13     </Grid>
PC14 </Page>

```

#### QUESTION 14

You need to enable the functionality to switch to the PictureChooserPage page.  
Which code segments should you insert at line CS42? (Each correct answer presents part of the solution. Choose all that apply.)

- A. 

```
this.Frame.Navigate(typeof(PictureChooserPage));
```
- B. 

```
this.Frame.GoForward(typeof(PictureChooserPage));
```
- C. 

```
private void OpenPicture_Click(object sender, RoutedEventArgs e)
```
- D. 

```
private void GetPictureButton_Click(object sender, RoutedEventArgs e)
```
- E. 

```
this.Frame.Navigate("PictureChooserPage");
```
- F. 

```
private void GetPicture_Click(object sender, RoutedEventArgs e)
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E
- F. Option F

**Correct Answer:** AD

**Explanation:**

From scenario:

You must perform the following tasks:

\* (D) Handle the Click event of the GetPictureButton control to switch from the current page to the PictureChooserPage page.

\* (A) After the user selects an image on the PictureChooserPage page, ensure that the app navigates back to the PictureSharerMainPage page.

#### QUESTION 15

You need to implement the required image magnification and manipulation requirements for selected images.

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

- A. Use the ManipulationDeltaRoutedEventArgs.Delta.Expansion value to implement optical zoom.
- B. Add a ManipulationMode attribute to the SelectedImage object and set the attribute to Translate.
- C. Use the ManipulationDeltaRoutedEventArgs.Delta.Scale value to implement optical zoom.
- D. Add a ManipulationDelta event to the SelectedImage object.
- E. Add a ManipulationMode attribute to the SelectedImage object and set the attribute to All.
- F. Add a ManipulationComplete event to the SelectedImage object.

**Correct Answer:** CDE

**Explanation:**

\* From scenario:

Ensure that users can change the magnification of the selected image and resize the image by using pinch and stretch gestures. Scaling should be fluid and precisely controlled by the user.

### QUESTION 16

You need to implement the OnNavigatedTo() method to meet the requirements. Which code segment should you insert at line CS13?

- A. 

```
if(e.NavigationMode == NavigationMode.Back)
{
    UpdateImage();
}
```
- B. 

```
if(e.TargetPageType == typeof(PictureSharerMainPage) && e.NavigationMode ==
NavigationMode.Back)
{
    UpdateImage();
}
```
- C. 

```
UpdateImage();
```
- D. 

```
if(e.SourcePageType == "PictureSharerMainPage"&& e.NavigationMode ==
NavigationMode.Back)
{
    UpdateImage();
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Correct Answer:** A

**Explanation:**

From scenario: Ensure that the OnNavigatedTo() method updates the current picture when a new picture is selected.

### QUESTION 17

You need to localize the Picture Sharer app in the required language.

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

all that apply.)

- A. Add a Uid attribute to any XAML elements that must be localized.
- B. Create a folder named fr-CA at the root of the project.
- C. Create a resource file named resources.res.
- D. Create a resource file named resources.resw.
- E. Create a folder named es-ES at the root of the project.
- F. Add a Name attribute to any XAML elements that must be localized.

**Correct Answer:** ABD

**Explanation:**

- \* (A) To localize a certain property of a XAML element you only need to add a x:Uid="SomeKey" attribute to the element and add the appropriate resource to the .resw file.
  - \* (B) The app must be localized for the French Canadian market.
  - \* (BD) Example: A French language resource named "Greeting" whose value is " Bonjour!". To create the resource file, add a folder named fr-FR to your project, and then add a resource file named Resources.resw to the folder.
  - \* In Windows Store apps, you designate the names of localized resource files by creating a folder to store the resources and images of a supported culture. You can then describe the resource by using the culture name (such as "ko-kr") followed by the default resource name and resource file extension (such as "ko-kr \Resources.resw").
- [http://msdn.microsoft.com/en-us/library/windows/apps/hh694557\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/windows/apps/hh694557(v=vs.110).aspx) URL:

**QUESTION 18**

You need to modify the GetContactsCompleted event handler to prevent the exception. Which code segment should you use to replace line CS32?

- A. 

```
Application.Current.Dispatcher.BeginInvoke(new Action(() => {  
    CoreDispatcherPriority.Background, RecipientsTextBlock.Text = "new value"; }));
```
- B. 

```
Action handler = delegate { RecipientsTextBlock.Text = email; };  
RecipientsTextBlock.Dispatcher.BeginInvoke(DispatcherPriority.Background, handler);
```
- C. 

```
InvokedHandler handler = delegate{ RecipientsTextBlock.Text = email; };  
Dispatcher.Invoke(CoreDispatcherPriority.Normal, handler, this, null);
```
- D. 

```
Action handler = delegate { RecipientsTextBlock.Text = email; };  
RecipientsTextBlock.Dispatcher.Invoke(DispatcherPriority.Normal, handler);
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Correct Answer:** C

**Explanation:**

- \* An exception is being thrown in the GetContactsCompleted event handler when the retrieved email address is assigned to the RecipientsTextBlock control. The exception message states: "The application called an interface that was marshalled for a different thread."

**QUESTION 19**

You need to retain the data that is entered in the TextBox control when the user navigates away from the PictureSharerMainPage page.

Which code segment should you insert at line CS08?



- A. `this.CaptionTextBox.NavigationCache = Windows.UI.Xaml.Navigation.NavigationCache.Enabled;`
- B. `this.NavigationCacheMode = Windows.UI.Xaml.Navigation.NavigationCacheMode.Required;`
- C. `this.CaptionTextBox.NavigationCacheMode = Windows.UI.Xaml.Navigation.NavigationCacheMode.Required;`
- D. `this.NavigationCache = Windows.UI.Xaml.Navigation.NavigationCache.Enabled;`

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Correct Answer:** B

**Explanation:**

\* NavigationCacheMode

Specifies caching characteristics for a page involved in a navigation.

\* NavigationCacheMode.Required

The page is cached and the cached instance is reused for every visit regardless of the cache size for the frame.

#### QUESTION 20

You need to configure the ContactPicker object.

Which code segment should you insert at line CS18?

- A. `picker.DesiredFields.Add(KnownContactField.LastName);`
- B. `picker.DesiredFields.Add("Email");`
- C. `picker.FiltersCollection.Add("LastName");`
- D. `picker.FiltersCollection.Add(KnownContactField.FirstName);`
- E. `picker.DesiredFields.Add(KnownContactField.Email);`

**Correct Answer:** E

**Explanation:**

The ContactPicker object must be filtered to display only email addresses.

## EnsurePass.com Members Features:

1. Verified Answers researched by industry experts.
2. Q&As are downloadable in PDF and VCE format.
3. 98% success Guarantee and **Money Back** Guarantee.
4. Free updates for **180** Days.
5. **Instant Access to download the Items**

View list of All Exam provided:

<http://www.ensurepass.com/certifications?index=A>

To purchase Lifetime Full Access Membership click here:

<http://www.ensurepass.com/user/register>

**Valid Discount Code for 2015: JREH-G1A8-XHC6**

To purchase the HOT Microsoft Exams:

<u>Microsoft</u>			
<a href="#">70-243</a>	<a href="#">70-347</a>	<a href="#">70-466</a>	<a href="#">70-515</a>
<a href="#">70-246</a>	<a href="#">70-410</a>	<a href="#">70-467</a>	<a href="#">70-516</a>
<a href="#">70-247</a>	<a href="#">70-411</a>	<a href="#">70-480</a>	<a href="#">70-519</a>
<a href="#">70-321</a>	<a href="#">70-412</a>	<a href="#">70-483</a>	<a href="#">70-583</a>
<a href="#">70-331</a>	<a href="#">70-413</a>	<a href="#">70-484</a>	<a href="#">70-640</a>
<a href="#">70-332</a>	<a href="#">70-414</a>	<a href="#">70-485</a>	<a href="#">70-649</a>
<a href="#">70-336</a>	<a href="#">70-417</a>	<a href="#">70-486</a>	<a href="#">70-668</a>
<a href="#">70-337</a>	<a href="#">70-461</a>	<a href="#">70-487</a>	<a href="#">70-680</a>
<a href="#">70-341</a>	<a href="#">70-462</a>	<a href="#">70-488</a>	<a href="#">70-687</a>
<a href="#">70-342</a>	<a href="#">70-463</a>	<a href="#">70-489</a>	<a href="#">70-688</a>
<a href="#">70-346</a>	<a href="#">70-464</a>	<a href="#">70-513</a>	<a href="#">70-689</a>



Guaranteed Success with EnsurePass VCE Software & PDF File