

Exam : 70-552(VB)

Title : UPGRADE:MCAD Skills to MCPD Wdws
Dvlpr by Using MS.NET Frmwk

Update : DEMO

<http://www.test4actual.com>

1. You are creating a Windows Form that includes a TextBox control named txtDate. When a user right-clicks within the text box, you want the application to display a MonthCalendar control. You need to implement a context menu that provides this functionality. What should you do?

A. Add the following code to the form initialization.
`Dim cal As New MonthCalendar()
Dim mnuContext As New ContextMenuStrip()
Dim host As New ToolStripControlHost(mnuContext)
txtDate.ContextMenuStrip = mnuContext`

B. Add the following code to the form initialization.
`Dim mnuContext As New ContextMenuStrip()
Dim cal As New MonthCalendar()
Dim host As`

`New ToolStripControlHost(cal)
mnuContext.Items.Add(host)
txtDate.ContextMenuStrip = mnuContext`

C. Add the following code to the form initialization.
`Dim ctr As New ToolStripContainer()
Dim cal As New MonthCalendar()
ctr.ContentPanel.Controls.Add(cal)
txtDate.Controls.Add(ctr)
Add a MouseClick event handler for the TextBox control that contains the following code.
If e.Button = MouseButton.Right Then
txtDate.Controls(0).Show()
End If`

D. Add a MouseClick event handler for the TextBox control that contains the following code.
If e.Button = MouseButton.Right Then
Dim mnuContext As New ContextMenuStrip()
Dim cal As New MonthCalendar()
Dim host As New ToolStripControlHost(cal)
mnuContext.Items.Add(host)
txtDate.ContextMenuStrip = mnuContext
End If

Answer: B

2. You are creating a Windows Forms application. You set the FlatAppearance.MouseOverBackColor property of a button to Blue. When testing the application, you notice that the background color does not change when you move the pointer over the button. You need to set the properties of the button so that the background color for the button changes to blue when the pointer moves over the button. What should you do?

A. Set the FlatStyle property to FlatStyle.Flat.

B. Set the FlatStyle property to FlatStyle.System.

C. Move the set statement for the FlatAppearance.MouseOverBackColor property to the Paint event.

D. Set the UseVisualStyleBackColor property to False.

Answer: A

3. You are customizing a Windows Form. You need to add an input control that provides AutoComplete suggestions to the user as the user types. Which two controls can you use to achieve this goal? (Each correct answer presents a complete solution. Choose two.)

A. TextBox control set to SingleLine mode

B. TextBox control set to MultiLine mode

C. ComboBox control

D. RichTextBox control

E. MaskedTextBox control

Answer: AC

4. You are creating a Windows Form. You add a TableLayoutPanel control named pnlLayout to the form. You set the properties of pnlLayout so that it will resize with the form. You need to create a three-column layout that has fixed left and right columns. The fixed columns must each remain 50 pixels wide when the form is resized. The middle column must fill the remainder of the form width when the form is resized. You add the three columns in the designer. Which code segment should you use to format the columns at run time?

- A. `pnlLayout.ColumnStyles.Clear()``pnlLayout.ColumnStyles.Add(New ColumnStyle(SizeType.Absolute, 50.0F))``pnlLayout.ColumnStyles.Add(New ColumnStyle(SizeType.AutoSize, 100.0F))``pnlLayout.ColumnStyles.Add(New ColumnStyle(SizeType.Absolute, 50.0F))`
- B. `pnlLayout.ColumnStyles(0).Width = 50.0F``pnlLayout.ColumnStyles(0).SizeType = SizeType.Absolute``pnlLayout.ColumnStyles(2).Width = 50.0F``pnlLayout.ColumnStyles(2).SizeType = SizeType.Absolute`
- C. `pnlLayout.ColumnStyles(0).Width = 50.0F``pnlLayout.ColumnStyles(0).SizeType = SizeType.Absolute``pnlLayout.ColumnStyles(1).Width = 100.0F``pnlLayout.ColumnStyles(1).SizeType = SizeType.AutoSize``pnlLayout.ColumnStyles(2).Width = 50.0F``pnlLayout.ColumnStyles(2).SizeType = SizeType.Absolute`
- D. `pnlLayout.ColumnStyles.Clear()``pnlLayout.ColumnStyles.Add(New ColumnStyle(SizeType.Absolute, 50.0F))``pnlLayout.ColumnStyles.Add(New ColumnStyle(SizeType.Percent, 100.0F))``pnlLayout.ColumnStyles.Add(New ColumnStyle(SizeType.Absolute, 50.0F))`

Answer: D

5. You are creating a Windows Form that contains several ToolStrip controls. You need to add functionality that allows a user to drag any ToolStrip control from one edge of the form to another. What should you do?

- A. Configure a ToolStripContainer control to fill the form. Add the ToolStrip controls to the ToolStripContainer control.
- B. Configure a Panel control to fill the form. Set the Anchor properties of the ToolStrip controls to Top, Bottom, Left, Right.
- C. Add the ToolStrip controls to another ToolStrip control that is hosted by a ToolStripControlHost control.
- D. Add the ToolStrip controls to the form. Set the Anchor properties of the ToolStrip controls to Top, Bottom, Left, Right. Set the FormBorderStyle property of the form to SizableToolWindow.

Answer: A

6. You are creating a Windows Forms application. You add an ErrorProvider component named erpErrors and a DateTimePicker control named dtpStartDate to the application. The application also contains other controls. You need to configure the application to display an error notification icon next to dtpStartDate when the user enters a date that is greater than today's date. Which two actions should you perform? (Each correct answer presents part of the solution. Choose two.)

- A. For the Validating event of `dateTimePicker1.StartDate`, create an event handler named `VerifyStartDate`.
- B. For the Validated event of `dateTimePicker1.StartDate`, create an event handler named `VerifyStartDate`.
- C. In the Properties Window for `dateTimePicker1.StartDate`, set the value of Error on `erpErrors` to Date out of range.
- D. In `VerifyStartDate`, call `erpErrors.SetError(dateTimePicker1.StartDate, "Date out of range")` if the value of `dateTimePicker1.StartDate.Value` is greater than today's date.
- E. In `VerifyStartDate`, call `erpErrors.SetError(dateTimePicker1.StartDate, null)` if the `dateTimePicker1.StartDate.Value` is greater than today's date.

Answer: AE

7. You are modifying a Windows Forms application. The application consists of a main window with many different controls. All of the controls provide tool tips that use the default `ToolTip` control settings. One group of controls provides tool tips that show regulatory guidance for the user. Users want the wait time when reading the tool tips and navigating among them to be minimal. You need to ensure that this group of controls provides short delays before the tool tips appear. What should you do?

- A. Set the `AutoPopDelay` property of the `ToolTip` control to 0 and the `InitialDelay` property to 100.
- B. Set the `AutomaticDelay` property of the `ToolTip` control to 0.
- C. Set the `InitialDelay` and `ReshowDelay` properties of the `ToolTip` control to 100.
- D. Set the `AutoPopDelay` property of the `ToolTip` control to 100.

Answer: C

8. You want to execute an event handler asynchronously from a Windows Form. You need to execute a method named `WorkHandler` by using an instance of the `BackgroundWorker` component named `bgwExecute`. Which two code segments should you use? (Each correct answer presents part of the solution. Choose two.)

- A. `Dim work As New EventHandler(AddressOf WorkHandler)`
- B. `Dim work As New ThreadStart(AddressOf WorkHandler)`
- C. `AddHandler bgwExecute.DoWork, AddressOf WorkHandler`
- D. `bgwExecute.RunWorkerAsync()`
- E. `bgwExecute.RunWorkerAsync(work)`

Answer: CD

9. You are customizing a Windows Form to use a `BackgroundWorker` component named `bgwExecute`. `bgwExecute` performs a database operation in an event handler named `WorkHandler`. You need to ensure that users can see the progress of the database operation by viewing a progress bar named `pbProgress`. You want the progress bar to appear when the database operation is 50 percent complete. Which code segment should you use?

- A.

```
Public Sub StartBackground()
    bgwExecute.WorkerReportsProgress = True
    AddHandler bgwExecute.ProgressChanged, AddressOf ProgressHandler
    bgwExecute.RunWorkerAsync()
End Sub
Sub WorkHandler(ByVal sender As Object, ByVal e As DoWorkEventArgs)
    bgwExecute.ReportProgress(50)
End Sub
Sub ProgressHandler(ByVal sender As Object, ByVal e As _
    ProgressChangedEventArgs)
    pbProgress.Value =
```

e.ProgressPercentageEnd Sub

```
B. Public Sub StartBackground()
    bgwExecute.WorkerReportsProgress = True
    AddHandler bgwExecute.ProgressChanged, AddressOf ProgressHandler
    Dim t As New ThreadStart(AddressOf WorkHandler)
    bgwExecute.RunWorkerAsync(t)End Sub
Sub WorkHandler()
    bgwExecute.ReportProgress(50)End Sub
Sub ProgressHandler(ByVal sender As Object, ByVal e As _
    ProgressChangedEventArgs)
    pbProgress.Value = e.ProgressPercentageEnd Sub
```

```
C. Public Sub StartBackground()
    bgwExecute.WorkerReportsProgress = True
    AddHandler bgwExecute.ProgressChanged, AddressOf ProgressHandler
    Dim t As New Thread(New ThreadStart(AddressOf
    WorkHandler))
    bgwExecute.RunWorkerAsync(t)End Sub
Sub WorkHandler()
    bgwExecute.ReportProgress(50)End Sub
Sub ProgressHandler(ByVal sender As Object, ByVal e As _
    ProgressChangedEventArgs)
    pbProgress.Value = e.ProgressPercentageEnd Sub
```

```
D. Public Sub StartBackground()
    bgwExecute.WorkerReportsProgress = True
    AddHandler bgwExecute.DoWork,
    AddressOf WorkHandler
    AddHandler bgwExecute.ProgressChanged,
    AddressOf ProgressHandler
    bgwExecute.RunWorkerAsync()End Sub
Sub WorkHandler(ByVal sender As Object, ByVal e As DoWorkEventArgs)
    bgwExecute.ReportProgress(50)End Sub
Sub ProgressHandler(ByVal sender As Object, ByVal e As _
    ProgressChangedEventArgs)
    pbProgress.Value = e.ProgressPercentageEnd Sub
```

Answer: D

10. You are customizing a Windows Form to update a database asynchronously in a method named WorkHandler. You need to ensure that the form displays a message box to the user that indicates the success or failure of the update. Which code segment should you use?

```
A. Private Sub StartBackgroundProcess()
    AddHandler bgwExecute.DoWork, AddressOf WorkHandler
    AddHandler
    bgwExecute.RunWorkerCompleted,
    AddressOf CompletedHandler
    bgwExecute.RunWorkerAsync()End Sub
Private Sub CompletedHandler(ByVal sender As Object, ByVal e As
    RunWorkerCompletedEventArgs)
    Dim result As Boolean = CType(e.Result, Boolean)
    If result = True Then
        MessageBox.Show("Update was successful")
    Else
        MessageBox.Show("Update failed")
    End IfEnd Sub
Private Sub WorkHandler(ByVal sender As Object, ByVal e As DoWorkEventArgs)
    '... e.Result = TrueEnd Sub
```

```
B. Private Sub StartBackgroundProcess()
    AddHandler bgwExecute.ProgressChanged, AddressOf
    CompletedHandler
    Dim tsBackground As New ThreadStart(AddressOf WorkHandler)
    bgwExecute.RunWorkerAsync(tsBackground)End Sub
Private Sub ProgressHandler(ByVal sender As Object, ByVal
    e As ProgressChangedEventArgs)
    Dim result As Boolean = CType(e.UserState, Boolean)
    If result = True Then
        MessageBox.Show("Update was successful")
    Else
        MessageBox.Show("Update failed")
    End IfEnd Sub
Private Sub WorkHandler()
    '... bgwExecute.ReportProgress(100, True)End Sub
```

```
C. Private Sub StartBackgroundProcess()
    AddHandler bgwExecute.RunWorkerCompleted, AddressOf
    CompletedHandler
    Dim tsBackground As New ThreadStart(AddressOf WorkHandler)
```

```
bgwExecute.RunWorkerAsync(tsBackground)End SubPrivate Sub CompletedHandler(ByVal sender As Object, ByVal e As RunWorkerCompletedEventArgs) Dim result As Boolean = CType(e.Result, Boolean) If result = True Then MessageBox.Show("Update was successful") Else MessageBox.Show("Update failed") End IfEnd SubPrivate Sub WorkHandler() '... bgwExecute.ReportProgress(100, True)End Sub
```

```
D. Private Sub StartBackgroundProcess() AddHandler bgwExecute.DoWork, AddressOf WorkHandler AddHandler bgwExecute.RunWorkerCompleted, AddressOf CompletedHandler bgwExecute.RunWorkerAsync()End SubPrivate Sub CompletedHandler(ByVal sender As Object, ByVal e As RunWorkerCompletedEventArgs) Dim result As Boolean = CType(e.Result, Boolean) If result = True Then MessageBox.Show("Update was successful") Else MessageBox.Show("Update failed") End IfEnd SubPrivate Sub WorkHandler(ByVal sender As Object, ByVal e As DoWorkEventArgs) '... bgwExecute.ReportProgress(100, True)End Sub
```

Answer: A

11. You want to execute an event handler asynchronously from a Windows Form. You need to write code that uses the BackgroundWorker component named bgwExecute to execute the WorkHandler method. Which code segment should you use?

- A. Dim work As New EventHandler(AddressOf WorkHandler)bgwExecute.RunWorkerAsync(work)
- B. Dim tsBackground As New ThreadStart(AddressOf WorkHandler)bgwExecute.ReportProgress(0, tsBackground)
- C. Dim tsBackground As New ThreadStart(AddressOf WorkHandler)bgwExecute.RunWorkerAsync(tsBackground)
- D. AddHandler bgwExecute.DoWork, AddressOf WorkHandlerbgwExecute.RunWorkerAsync()

Answer: D

12. You are customizing a Windows Form to update a database asynchronously by using an instance of a BackgroundWorker component named bgwExecute. You start the component by using the following code.

```
Private Sub StartBackgroundProcess()  
    AddHandler bgwExecute.DoWork, _  
        New DoWorkEventHandler(AddressOf WorkHandler)  
    AddHandler bgwExecute.RunWorkerCompleted, _  
        New RunWorkerCompletedEventHandler(AddressOf _  
            CompletedHandler)  
    AddHandler bgwExecute.ProgressChanged, _  
        New ProgressChangedEventHandler(AddressOf ProgressChanged)  
    bgwExecute.RunWorkerAsync()  
End Sub
```

If the UpdateDB method that is called by the BackgroundWorker component returns the value False, you need to display a message box to the user that indicates that the update failed. Which code segment should you use?

- A. Sub WorkHandler(ByVal sender As Object, ByVal e As DoWorkEventArgs) If Not UpdateDB() Then

```
MessageBox.Show("Update failed") End IfEnd Sub
```

```
B. Sub CompletedHandler(ByVal sender As Object, ByVal e As _ RunWorkerCompletedEventArgs) If Not UpdateDB() Then MessageBox.Show("Update failed") End IfEnd Sub
```

```
C. Sub WorkHandler(ByVal sender As Object, ByVal e As DoWorkEventArgs) e.Result = UpdateDB()End Sub Sub CompletedHandler(ByVal sender As Object, ByVal e As _ RunWorkerCompletedEventArgs) If Not CBool(e.Result) Then MessageBox.Show("Update failed") End IfEnd Sub
```

```
D. Sub WorkHandler(ByVal sender As Object, ByVal e As DoWorkEventArgs) e.Result = UpdateDB()End Sub Sub CompletedHandler(ByVal sender As Object, ByVal e As _ RunWorkerCompletedEventArgs) If Not CBool(e.Result) Then bgwExecute.ReportProgress(0) End IfEnd Sub Sub ProgressChanged(ByVal sender As Object, ByVal e As _ ProgressChangedEventArgs) If e.ProgressPercentage = 0 Then MessageBox.Show("Update failed") End IfEnd Sub
```

Answer: C

13. You are customizing a Windows Form to asynchronously update a database. You need to ensure that the form displays a message box to the user that indicates the success or failure of the update. Which three code segments should you use? (Each correct answer presents part of the solution. Choose three.)

```
A. Private Sub StartBackgroundProcess() AddHandler bgwExecute.DoWork, AddressOf WorkHandler AddHandler bgwExecute.RunWorkerCompleted, AddressOf CompletedHandler bgwExecute.RunWorkerAsync()End Sub
```

```
B. Private Sub StartBackgroundProcess() AddHandler bgwExecute.ProgressChanged, AddressOf CompletedHandler Dim tsBackground As New ThreadStart(AddressOf WorkHandler) bgwExecute.RunWorkerAsync(tsBackground)End Sub
```

```
C. Private Sub StartBackgroundProcess() AddHandler bgwExecute.RunWorkerCompleted, AddressOf CompletedHandler Dim tsBackground As New ThreadStart(AddressOf WorkHandler) bgwExecute.RunWorkerAsync(tsBackground)End Sub
```

```
D. Sub WorkHandler(ByVal sender As Object, ByVal e As DoWorkEventArgs) ... e.Result = TrueEnd Sub
```

```
E. Sub WorkHandler(ByVal sender As Object, ByVal e As DoWorkEventArgs) ... bgwExecute.ReportProgress(100, True)End Sub
```

```
F. Sub CompletedHandler(ByVal sender As Object, ByVal e As _ RunWorkerCompletedEventArgs) Dim result As Boolean = CBool(e.Result) If result Then MessageBox.Show("Update was successful") Else MessageBox.Show("Update failed") End IfEnd Sub
```

```
G. Sub ProgressHandler(ByVal sender As Object, ByVal e As _ ProgressChangedEventArgs) Dim result As Boolean = CBool(e.UserState) If result Then MessageBox.Show("Update was successful") Else MessageBox.Show("Update failed") End IfEnd Sub
```

Answer: ADF

14. You are creating an application named App1. You use ClickOnce deployment to distribute App1.exe and multiple assemblies. Some users require only some of the functionality in App1. You need to limit the size of the initial download of the application. You also need to ensure that users can download the assemblies on demand. Which three actions should you perform? (Each correct answer presents part of the solution. Choose three.)

- A. Mark each dependency in App1.exe.manifest as optional.
- B. Mark each dependency in App1.application as optional.
- C. Create an event handler for the AppDomain.ResourceResolve event named ResolveAssembly.
- D. Create an event handler for the AppDomain.AssemblyLoad event named ResolveAssembly.
- E. In the ResolveAssembly event handler, set the ApplicationDeployment.CurrentDeployment.ActivationUri property to the location of your required assembly.
- F. In the ResolveAssembly event handler, call ApplicationDeployment.DownloadFiles and pass in the name of the assembly you want.

Answer: ACF

15. You create a Windows-based application that requires the use of a COM component. You need to create a ClickOnce deployment package to distribute the application from an Internet Web site. Which two actions should you perform? (Each correct answer presents part of the solution. Choose two.)

- A. Set the Isolated property of the COM component references in the application project to False.
- B. Set the Isolated property of the COM component references in the application project to True.
- C. Verify that the user is using Microsoft Windows XP.
- D. Verify that the user is using Microsoft Windows 2000.
- E. Assign RegistryPermission to the application.

Answer: BC

16. You are configuring a ClickOnce deployment that allows users to install your application from the Internet zone under partial trust permissions. You want the application to access data that resides on the same remote server from which the application is installed. You need to add one or more types of data access that are allowed under partial trust permissions to your application. Which type or types of data access are allowed? (Choose all that apply.)

- A. data access through HTTP with System.Net.WebClient
- B. data access through XML Web services
- C. data access through System.Data.SqlClient
- D. data access through HTTP with System.Net.HttpWebRequest

Answer: ABD

17. A Windows Forms application reads the following XML file.

```
<?xml version="1.0"?>
<x:catalog xmlns:x="urn:books">
```

```
<book id="bk101">
  <author>Gambardella, Matthew</author>
  <title>XML Developer's Guide</title>
</book>
<book id="bk102">
  <author>Ralls, Kim</author>
  <title>Midnight Rain</title>
</book>
</x:catalog>
```

The form initialization loads this file into an XmlDocument object named docBooks. You need to populate a ListBox control named lstBooks with the concatenated book ID and title of each book. Which code segment should you use?

- A. Dim elements As XmlNodeList = docBooks.GetElementsByTagName("book")Dim node As XmlElementFor Each node In elements Dim s As String = node.GetAttribute("id") + " - " s = s + node.SelectSingleNode("title").InnerText lstBooks.Items.Add(s)Next node
- B. Dim elements As XmlNodeList = docBooks.GetElementsByTagName("book")Dim node As XmlElementFor Each node In elements Dim s As String = node.SelectSingleNode("id").ToString() + " - " s = s + node.GetAttribute("title") lstBooks.Items.Add(s)Next node
- C. Dim elements As XmlNodeList = docBooks.GetElementsByTagName("book")Dim node As XmlElementFor Each node In elements Dim s As String = node.GetAttribute("id") + " - " s = s + node.SelectSingleNode("title").Value lstBooks.Items.Add(s)Next node
- D. Dim elements As XmlNodeList = docBooks.GetElementsByTagName("book")Dim node As XmlElementFor Each node In elements lstBooks.Items.Add(node.InnerXml)Next node

Answer: A

18. A Windows Forms application contains the following code segment.

```
Dim SQL As String = "SELECT EmployeeID, LastName, FirstName FROM Employees"
Dim da As New SqlDataAdapter(SQL, connStr)
Dim dt As New DataTable()
da.MissingSchemaAction = MissingSchemaAction.AddWithKey
Dim bld As New SqlCommandBuilder(da)
da.Fill(dt)
```

The application allows the user to add rows to the data table. The application will propagate these additions to the database. If the addition of any row fails, the other rows must still be added. The code must log how many new rows failed to be added. You need to propagate the additions to the database and log a failed count. Which code segment should you use?

- A. da.ContinueUpdateOnError = Trueda.Update(dt)Dim dtErrors As DataTable =
dt.GetChanges(DataRowState.Unchanged)Trace.WriteLine((dtErrors.Rows.Count.ToString() + " rows not added."))
- B. da.ContinueUpdateOnError = Falseda.Update(dt)Dim dtErrors As DataTable =
dt.GetChanges(DataRowState.Unchanged)Trace.WriteLine((dtErrors.Rows.Count.ToString() + " rows not added."))
- C. da.ContinueUpdateOnError = Trueda.Update(dt)Dim rows As DataRow() =
dt.GetErrors()Trace.WriteLine((rows.Length.ToString() + " rows not added."))
- D. da.ContinueUpdateOnError = Falseda.Update(dt)Dim rows As DataRow() =
dt.GetErrors()Trace.WriteLine((rows.Length.ToString() + " rows not added."))

Answer: C

19. You are creating a Windows Forms application. The application loads a data table named dt from a database and modifies each value in the data table.

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

```
01 Dim row As DataRow
02 For Each row In dt.Rows
03     Dim col As DataColumn
04     For Each col In dt.Columns
05
06         Trace.WriteLine(str)
07     Next col
08 Next row
```

You need to format the string named str to show the value of the column at the time the data is loaded and the current value in the column. Which code segment should you add at line 05?

- A. Dim str As String = String.Format("Column was {0} is now {1}", row(col), row(col, DataRowVersion.Current))
- B. Dim str As String = String.Format("Column was {0} is now {1}", row(col, DataRowVersion.Default), row(col))
- C. Dim str As String = String.Format("Column was {0} is now {1}", row(col), row(col, DataRowVersion.Proposed))
- D. Dim str As String = String.Format("Column was {0} is now {1}", row(col, DataRowVersion.Original), row(col))

Answer: D

20. A Windows Forms application loads an XmlDocument from a file named books.xml. You need to validate the XML against a schema that is contained in the books.xsd file when the XML loads. What should you do?

- A. Associate the schema file with an XmlReader.Load the XmlDocument by using the XmlReader.
- B. Add the schema to the Schemas property of the XmlDocument.Call the Load method of the XmlDocument by setting the filename parameter to books.xsd.
- C. Call the Load method of the XmlDocument by setting the filename parameter to books.xsd, and then call the Load method by setting the filename parameter to books.xml.

D. Call the Load method of the XmlDocument by setting the filename parameter to books.xsd. Programmatically add the attribute xsi:schemaLocation to the root node. Set the value of this attribute to books.xsd.

Answer: A

www.test4actual.com