70-504

TS: MS.NET Framework 3.5 Workflow Foundation Application Developer

Q&A

DEMO Version

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.

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.

Microsoft 70-504 (VB)

Question: 1
You create a Windows Workflow Foundation application by using Microsoft .NET Framework 3.5. The application contains a workflow named AdmitWorkflow in the namespace Hospital.Patient. The application uses strong-named assemblies. You plan to create an administrative application to monitor the workflow database. The administrative application must return a list of idle AdmitWorkflow workflows. You need to correctly configure the SqlTrackingQueryOptions class. Which code segment should you use?

A. Dim options As New SqlTrackingQueryOptions()options.WorkflowStatus =
WorkflowStatus.Runningoptions.WorkflowType = _
Type.GetType("Hospital.Patient.AdmitWorkflow")
B. Dim options As New SqlTrackingQueryOptions()options.WorkflowStatus =
WorkflowStatus.Suspendedoptions.WorkflowType = _
Type.GetType("Hospital.Patient.AdmitWorkflow")
C. Dim options As New SqlTrackingQueryOptions()options.WorkflowStatus =
WorkflowStatus.Runningoptions.WorkflowType = _
Type.GetType("Hospital.Patient.AdmitWorkflow," + _ "
Hospital.Patient, Version=1.0.0.0, Cult0ure=neutral," + _ "
PublicKeyToken=0123456789ABCDEF")
D. Dim options As New SqlTrackingQueryOptions()options.WorkflowStatus =
WorkflowStatus.Suspendedoptions.WorkflowType = _
Type.GetType("Hospital.Patient.AdmitWorkflow," + _
" Hospital.Patient, Version=1.0.0.0, Culture=neutral," + _ "
PublicKeyToken=0123456789ABCDEF")

Answer: C

Question: 2
You create a Windows Workflow Foundation application by using Microsoft .NET Framework 3.5. The application uses state machineCbased workflows. As the workflow progresses, each state requires the name of the previous state. The workflow must be able to return the previous state at any point during the workflow. You need to create a method that returns the name of the last state. Which code segment should you use?

A. Dim wi As New StateMachineWorkflowInstance(runtime, workflowId)Return wi.StateHistory(0)
B. Dim wi As New StateMachineWorkflowInstance(runtime, workflowId)Return
wi.CurrentState.Parent.Name
C. Dim wi As New StateMachineWorkflowInstance(runtime, workflowId)Return
wi.States(wi.States.Count - 1).Name
D. Dim wi As New StateMachineWorkflowInstance(runtime, workflowId)Return
wi.StateHistory(wi.StateHistory.Count - 1)

Answer: D

Question: 3
You are creating a Windows Workflow Foundation application by using Microsoft .NET Framework 3.5. You need to ensure that the application records event tracking information in the Windows Event Log. What should you do?

A. Derive one custom class each from the TrackingService class and the TrackingChannel class. Return the custom class derived from the GetTrackingChannel method of the TrackingChannel class. Write the tracking information to the Windows Event Log in the Send method.
B. Derive one custom class each from the TrackingService class and the TrackingChannel class. Return the custom class derived from the GetTrackingChannel method of the TrackingChannel class. Write the tracking information to the Windows Event Log in the GetProfile method.
C. Derive one custom class each from the TrackingService class and the TrackingProfile class. Return the custom class derived from the TrackingProfile from the GetProfile method. Write the tracking information to the Windows Event Log in the constructor of the custom class derived from the TrackingProfile class.
D. Derive one custom class each from the TrackingChannel class and the TrackingProfile class. Create an instance of the custom derived TrackingProfile class in the Send method. Write the tracking information to the Windows Event Log in the constructor of the custom class derived from the TrackingProfile class.

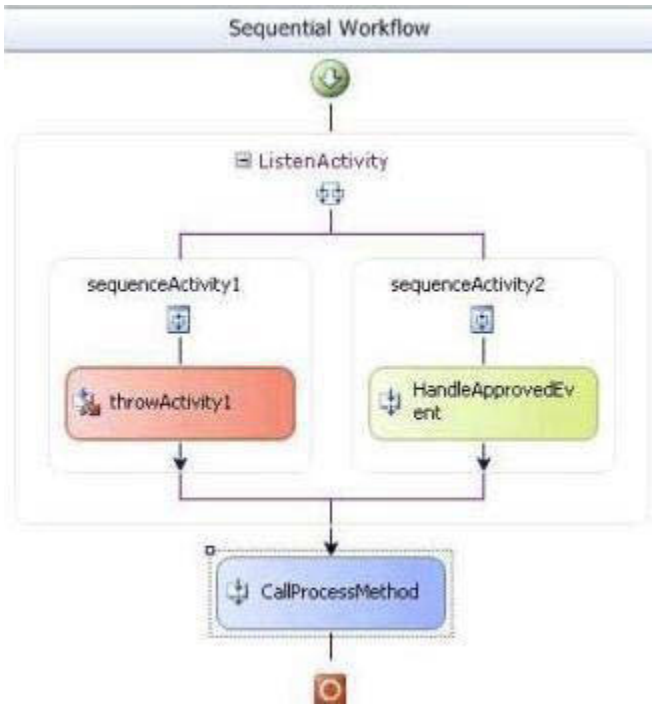
Answer: A

Question: 4
You are creating a Windows Workflow Foundation application by using Microsoft .NET Framework 3.5. The application uses a state machineCbased workflow that takes 10 to 15 days to complete. The workflow will be persisted when idle. The workflow communicates with a custom class that implements the IWorkflowAction interface. The interface contains events that the workflow will handle. The events require a custom EventArgs class. You need to implement the WorkflowActionEventArgs class. Which code segment should you use?

A. Public Class WorkflowActionEventArgs Inherits EventArgs Public InstanceId As Guid Public Action As
String Public Sub New(ByVal instanceId As Guid, ByVal action As String) Me.InstanceId = instanceId
Me.Action = action End SubEnd Class
B. <Serializable()> _Public Class WorkflowActionEventArgs Inherits EventArgs Public InstanceId As Guid
Public Action As String Public Sub New(ByVal instanceId As Guid, ByVal action As String) Me.InstanceId
= instanceId Me.Action = action End SubEnd Class
C. Public Class WorkflowActionEventArgs Inherits ExternalDataEventArgs Public Action As String Public
Sub New(ByVal instanceId As Guid, ByVal action As String) MyBase.New(instanceId) Me.Action = action
End SubEnd Class
D. <Serializable()> _Public Class WorkflowActionEventArgs Inherits ExternalDataEventArgs Public Action
As String Public Sub New(ByVal instanceId As Guid, ByVal action As String) MyBase.New(instanceId)
Me.Action = action End SubEnd Class

Answer: D

Question: 5
You create a Windows Workflow Foundation application by using Microsoft .NET Framework 3.5. The workflow design is as shown in the following exhibit. (Click the Exhibit button.)



You need to add an activity before the throwActivity1 activity. You also need to ensure that the added activity allows the throwActivity1 activity to throw an exception only if the Approved event is not received in four hours. What should you do?

A. Add a DelayActivity activity and set the TimeoutDuration property to four hours.
B. Add a CodeActivity activity. In the CodeActivity activity, call the Thread.Sleep method. Pass a time span of four hours to the Thread.Sleep method.
C. Add a CodeActivity activity. In the CodeActivity activity, instantiate a Timer class. Set the Interval property of the Timer class to four hours. Handle the Elapsed event and check if the event has been raised.
D. Add a WhileActivity activity. In the WhileActivity activity, add a SuspendActivity activity. In the Condition property of the WhileActivity activity, create a code condition and attach a delegate to the Approved event in the workflow.

Answer: A

Question: 6
You use a built-in tracking service to track specific workflow parameters. You need to check whether the workflow parameters have been stored in the tracking database. What should you do? (Each correct answer presents part of a solution. Choose two.)

A. Display the contents of the WorkflowInstance table of the tracking database.
B. Include the SqlTrackingQuery class in a code segment to retrieve tracked workflows and SqlTrackingWorkflowInstance class to inspect them.
C. Use the ActivityTrackingLocation class to determine if the value has been set to a database.
D. Display the contents of the TrackingDataItem table of the tracking database.

Answer: B, D

Question: 7
You create a Windows Workflow Foundation application by using Microsoft .NET Framework 3.5. The application uses a sequential workflow. The workflow calls an external method to notify a list

of users to carry out tasks. The list of users varies in size and composition from one workflow instance to another. The list is implemented as a string array. When a user completes a task, the host application raises a TaskCompleted event. You need to ensure that the users receive their notifications simultaneously. What should you do?

A. Add the CallExternalMethodActivity and the HandleExternalEventActivity activities in a While activity. Set the While activity to loop through the entire string array.
B. Add the CallExternalMethodActivity and the HandleExternalEventActivity activities in a Replicator activity. Set the ExecutionType property of the Replicator activity to Parallel.
C. Add the CallExternalMethodActivity and the HandleExternalEventActivity activities in a Replicator activity. Set the ExecutionType property of the Replicator activity to Sequence.
D. Add a ParallelActivity activity to the workflow. Add branches to the activity such that the number of branches is equal to the number of persons to be notified. Add the CallExternalMethodActivity and the HandleExternalEventActivity activities to each branch.

Answer: B

Question: 8
You create a Windows Workflow Foundation application by using Microsoft .NET Framework 3.5. The application uses a sequential workflow. The workflow is implemented in a class named ProcessOrders. The workflow contains a dependency property named EmployeeID. You need to ensure that the EmployeeID property is assigned a value when the host application tries to create a new workflow instance. Which code segment should you use?

A. Dim runtime As New WorkflowRuntime()Dim processOrders As New ProcessOrders()processOrders.EmployeeID = "NBK"Dim instance As WorkflowInstance = _ runtime.CreateWorkflow(GetType(ProcessOrders))
B. Dim runtime As New WorkflowRuntime()Dim processOrders As New ProcessOrders()processOrders.SetValue(_ processOrders.EmployeeIDProperty, "NBK")Dim instance As WorkflowInstance = _ runtime.CreateWorkflow(GetType(ProcessOrders))
C. Dim runtime As New WorkflowRuntime()Dim dict As Dictionary(Of String, Object) = _ New Dictionary(Of String, Object)()dict.Add("EmployeeID", "NBK")Dim instance As WorkflowInstance = _ runtime.CreateWorkflow(GetType(ProcessOrders), dict)
D. Dim runtime As New WorkflowRuntime()Dim dict As Dictionary(Of String, Object) = _ New Dictionary(Of String, Object)()dict.Add("EmployeeIDProperty", "NBK") Dim instance As WorkflowInstance = _runtime.CreateWorkflow(GetType(ProcessOrders), dict)

Answer: C

Question: 9
You create a Windows Workflow Foundation application by using Microsoft .NET Framework 3.5. The application contains a state workflow. You write the following code segment. Dim amount As Integer = 10 Dim runtime As New WorkflowRuntime () Dim instance As WorkflowInstance = _ runtime.CreateWorkflow (GetType (DynamicUpdateWorkflow)) instance.Start () Dim smwi As New StateMachineWorkflowInstance (runtime, _ instance.InstanceId)
A dependency property named Status is defined in this workflow.
The value of a variable named amount is used to set the state of the workflow.
You need to ensure that the host application changes the state of the workflow on the basis of the value of the amount variable.
What are the two possible code segments that you can use to achieve this goal? (Each correct answer presents a complete solution. Choose two.)

A. If amount >= 1000 Then smwi.SetState("HighValueState")Else
smwi.SetState("LowValueState")End If
B. If amount >= 1000 Then smwi.StateMachineWorkflow.SetValue _
(DynamicUpdateWorkflow.StatusProperty, "HighValueState")Else
smwi.StateMachineWorkflow.SetValue _
(DynamicUpdateWorkflow.StatusProperty, "LowValueState")End If
C. If amount >= 1000 Then instance.GetWorkflowDefinition().SetValue
(DynamicUpdateWorkflow.StatusProperty, "HighValueState")Else
instance.GetWorkflowDefinition().SetValue (DynamicUpdateWorkflow.StatusProperty,
"LowValueState")End If
D. If amount >= 1000 Then Dim high As StateActivity = _
CType(smwi.StateMachineWorkflow.Activities("HighValueState"), _ StateActivity)
smwi.SetState(high)Else
Dim low As StateActivity = _ CType(smwi.StateMachineWorkflow.Activities("LowValueState"),
_
StateActivity) smwi.SetState(low)End If

Answer: A, D

Question: 10
A custom activity defined in an assembly named LitwareActivities is defined as follows:
Namespace LitwareActivities
Public Class WriteLineActivity
Inherits Activity
Protected Overrides Function Execute(ByVal executionContext As
System.Workflow.ComponentModel.ActivityExecutionContext) _ As
System.Workflow.ComponentModel.ActivityExecutionStatus
Console.WriteLine(Message)
Return ActivityExecutionStatus.Closed
End Function
Private aMessage As String
Public Property Message() As String
Get
Return aMessage
End Get
Set(ByVal value As String)
aMessage = value
End Set
End Property
End Class
End Namespace
You need to create a sequential workflow where the execution path can be generated on the fly by an application. Which XML code segment should you use?

A. <SequentialWorkflowActivity

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/workflow"xmlns:x="http://schemas.microsoft.com/winfx/2006/
xmlns:Litware="clr-
namespace:LitwareActivities;assembly=LitwareActivities"><Litware:WriteLineActivity
Message="Hello, WF"/></SequentialWorkflowActivity>
B. <Workflow

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

xmlns:Litware="clr-
namespace:LitwareActivities;assembly=LitwareActivities"><Litware:WriteLineActivity
Message="Hello, WF"/></Workflow>

C. <Workflow xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
class:Litware="clr-namespace:LitwareActivities;assembly=LitwareActivities">
<Litware:WriteLineActivity
Message="Hello, WF"/></Workflow>

D. <SequentialWorkflowActivity
class:Litware="clr-namespace:LitwareActivities;assembly=LitwareActivities">
<Litware:WriteLineActivity
Message="Hello, WF"/></SequentialWorkflowActivity>

Answer: A