



**Vendor:** Microsoft

**Exam Code:** 70-504

**Exam Name:** TS: Microsoft .NET Framework 3.5 -  
Workflow

**Version:** DEMO

1: You create a workflow host application by using Microsoft .NET Framework 3.5. You use Windows Workflow Foundation to create the application. You need to configure the workflow runtime to ensure that all the workflow instances run asynchronously. Which code segment should you use?

A. `WorkflowRuntime runtime = new`

```
WorkflowRuntime();runtime.StartRuntime();WorkflowInstance instance =  
runtime.CreateWorkflow(typeof(CustomerWorkflow));instance.Start();
```

B. `WorkflowRuntime runtime = new`

```
WorkflowRuntime();runtime.StartRuntime();ManualWorkflowSchedulerService scheduler =  
runtime.GetService<ManualWorkflowSchedulerService>();WorkflowInstance instance =  
runtime.CreateWorkflow(typeof(CustomerWorkflow));scheduler.RunWorkflow(instance.InstanceId);
```

C. `WorkflowRuntime runtime = new WorkflowRuntime();ManualWorkflowSchedulerService scheduler = new`

```
ManualWorkflowSchedulerService();runtime.AddService(scheduler);runtime.StartRuntime();WorkflowInstance instance =  
runtime.CreateWorkflow(typeof(CustomerWorkflow));instance.Start();
```

D. `WorkflowRuntime runtime = new`

```
WorkflowRuntime();runtime.StartRuntime();DefaultWorkflowSchedulerService scheduler =  
new DefaultWorkflowSchedulerService();runtime.AddService(scheduler);WorkflowInstance instance =  
runtime.CreateWorkflow(typeof(CustomerWorkflow));instance.Start();
```

**Correct Answers: A**

2: You create a Windows Workflow Foundation application by using Microsoft .NET Framework 3.5.

The application is exposed as a Web service. You use `WebServiceInputActivity` activities in your workflow.

You need to ensure that the following requirements are met:

- All exceptions are caught at runtime.

- The exceptions are thrown as SOAP exceptions back to the client application without changing the course of the workflow.

What should you do?

A. Add the `ThrowActivity` activity after the `WebServiceInputActivity` activity.

B. Add the `ThrowActivity` activity before the `WebServiceInputActivity` activity.

C. Add the `WebServiceFaultActivity` activity after the `WebServiceInputActivity` activity.

D. Add the `WebServiceFaultActivity` activity before the `WebServiceInputActivity` activity.

**Correct Answers: C**

3: You create a Windows Workflow Foundation application by using Microsoft .NET Framework 3.5.

The application has a workflow named `WfService`.

You write the following code segment.

```
public interface MyService  
{  
    void DoWorkflow(int val1,ref int val2);
```

}

You need to expose the DoWorkflow method as a Web service.

What should you do?

A. Add a WebServiceInputActivity activity. Bind val1 and val2 as parameters.

B. Add a WebServiceInputActivity activity. Bind val2 as a parameter.

Add a WebServiceOutputActivity activity. Bind val1 as a return value.

C. Add a WebServiceInputActivity activity. Bind val1 as a parameter.

Add a WebServiceOutputActivity activity. Bind val2 as a return value.

D. Add a WebServiceInputActivity activity. Bind val1 and val2 as parameters.

Add a WebServiceOutputActivity activity. Bind val2 as a return value.

**Correct Answers: D**

4: You are creating a Windows Workflow Foundation custom activity by using Microsoft .NET Framework 3.5.

The custom activity has the following features:

It uses a voting process.

It completes execution after the receipt of a Yes or a No vote from an end user.

The voting process is managed by a local service of type VotingService. The voting process can take 15 to 20 days.

You need to ensure that the VotingService service informs the custom activity when a vote is received.

What should you do?

A. Implement the VotingService service to invoke a method in the custom activity and pass the voting data as a workflow parameter.

B. Implement the VotingService service to schedule the custom activity to execute and pass the voting data as a workflow parameter.

C. Implement the VotingService service to enqueue the voting data in a workflow queue that was configured by the custom activity.

D. Create a dependency property in the custom activity for the voting data. Implement the VotingService service to configure the dependency property when data is received.

**Correct Answers: C**

5: You create a Windows Workflow Foundation application by using Microsoft .NET Framework 3.5.

You add a class to the workflow that implements the IPendingWork interface. The class interacts with a Microsoft SQL Server 2005 database.

All database write operations must be performed in a transaction. You implement a Commit method of the IPendingWork interface.

You need to complete the Commit method.

Which code segment should you use?

A. `public void Commit(Transaction trans, ICollection items) { SqlConnection con = new SqlConnection("<con str>"); con.BeginTransaction(); ...}`

B. `public void Commit(Transaction trans, ICollection items) { SqlConnection con = new SqlConnection("<con str>"); con.EnlistTransaction(trans); ...}`

```
C.public void Commit(Transaction trans, ICollection items) { SqlConnection con = new SqlConnection("<con str>"); con.EnlistTransaction(new CommittableTransaction()); ...}
```

```
D.public void Commit(Transaction trans, ICollection items) { using (TransactionScope tscope=new TransactionScope()) { SqlConnection con = new SqlConnection("<con str>"); ... tscope.Complete(); }}
```

**Correct Answers: B**

6: You create a Windows Workflow Foundation application by using Microsoft .NET Framework 3.5. The application is a high-volume shopping cart system that processes over 100,000 transactions every day. The application tracks workflows by using the `SqlTrackingService` class. The application must partition data based on a 24-hour period. You need to configure the tracking service. What should you do?

A. In the `app.config` file, set the `PartitionOnCompletion` attribute of the `SqlTrackingService` class to `true`. Execute the stored procedure named `dbo.SetPartitionInterval` by passing the parameter value `m`.

B. In the `app.config` file, set the `PartitionOnCompletion` attribute of the `SqlTrackingService` class to `false`. Execute the stored procedure named `dbo.PartitionCompletedWorkflowInstances`.

C. In the code segment used to create the `SqlTrackingService` class, set the `PartitionOnCompletion` property to `true`. Execute the stored procedure named `dbo.PartitionCompletedWorkflowInstances`.

D. In the code segment used to create the `SqlTrackingService` class, set the `PartitionOnCompletion` property to `true`. Execute the stored procedure named `dbo.SetPartitionInterval` by passing the parameter value `d`.

**Correct Answers: D**

7: You create a Windows Workflow Foundation application by using Microsoft .NET Framework 3.5. When a user logs in, the application uses the `InvokeWebServiceActivity` activity to retrieve data from a Web service. You need to retrieve the logged-in user details from a local database when the Web service is being called. Which code segment should you use?

```
A.this.InvokeWebServiceActivity1.OnInvoked += new  
EventHandler<InvokeWebServiceEventArgs>(OnInvoked); void OnInvoked(object sender,  
InvokeWebServiceEventArgs e){ ...}
```

```
B.this.InvokeWebServiceActivity1.OnInvoking += new  
EventHandler<InvokeWebServiceEventArgs>(OnInvoking); void OnInvoking(object sender,  
InvokeWebServiceEventArgs e){ ...}
```

```
C.this.InvokeWebServiceActivity1.OnStatusChanged += new  
EventHandler<ActivityExecutionStatusChangedEventArgs> (OnStatusChanged); private void  
OnStatusChanged(object sender, EventArgs e){ if (InvokeWebServiceActivity1.ExecutionStatus  
== ActivityExecutionStatus.Executing) { ... }}
```

```
D.this.InvokeWebServiceActivity1.OnStatusChanged += new  
EventHandler<ActivityExecutionStatusChangedEventArgs> (OnStatusChanged); private void  
OnStatusChanged(object sender, EventArgs e){ if (InvokeWebServiceActivity1.ExecutionStatus  
== ActivityExecutionStatus.Compensating) { ... }}
```

**Correct Answers: B**

8: 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.

**Correct Answers: A**

9: You create a Windows Workflow Foundation application by using Microsoft .NET Framework 3.5. The application uses a sequential workflow. The application must listen to the Approved and the Rejected events that are raised by the host. You need to ensure that the workflow waits for one of the events to be raised before it completes execution. What should you do?

A. Add a ParallelActivity activity that has two branches to the workflow. Add a HandleExternalEventActivity activity to each branch. Configure one branch to handle the Approved event and the other one to handle the Rejected event.

B. Add a ListenActivity activity that has two branches to the workflow. Add a HandleExternalEventActivity activity to each branch. Configure one HandleExternalEventActivity activity to handle the Approved event and the other one to handle the Rejected event.

C. Add a ReplicatorActivity activity that has the ExecutionType property set to Parallel. Add two HandleExternalEventActivity activities to the ReplicatorActivity activity. Configure one HandleExternalEventActivity activity to handle the Approved event and the other one to handle the Rejected event.

D. Add a ParallelActivity activity that has two branches to the workflow. Add an EventHandlingScopeActivity activity to each branch. Add a HandleExternalEventActivity activity to each EventHandlingScopeActivity activity. Configure one HandleExternalEventActivity activity to handle the Approved event and the other one to handle the Rejected event.

**Correct Answers: B**

10: You are creating a Windows Workflow Foundation application by using Microsoft .NET Framework 3.5. The application contains a workflow that performs two tasks. The first task will complete before the second task begins. You need to ensure that corrective action is taken on work performed by the first task if an exception occurs while processing the second task. What should you do?

A. Place the activities that perform the first task in a `CompensatableSequenceActivity` activity. Add a compensation handler to the `CompensatableSequenceActivity` activity. Add an activity at the workflow level to perform the second task. Add an exception handler that traps the `System.Exception` exception and contains a `CompensateActivity` activity with a `TargetPropertyName` property that points to the `CompensatableSequenceActivity` activity.

B. Place the activities that perform the first task in a `CompensatableSequenceActivity` activity. Add a cancellation handler to the `CompensatableSequenceActivity` activity. Add an activity at the workflow level to perform the second task. Add an exception handler that traps the `System.Exception` exception and contains a `CompensateActivity` activity with a `TargetPropertyName` property that points to the `CompensatableSequenceActivity` activity.

C. Place the activities that perform the first task in a `CompensatableSequenceActivity` activity. Add a fault handler to the `CompensatableSequenceActivity` activity. Add an activity at the workflow level to perform the second task. Add an exception handler that traps the `System.Exception` exception and contains a `CompensateActivity` activity with a `TargetPropertyName` property that points to the `CompensatableSequenceActivity` activity.

D. Place the activities that perform the first task in a `CompensatableSequenceActivity` activity. Add a fault handler to the `CompensatableSequenceActivity` activity. Add an activity at the workflow level to perform the second task and enclose it in a `CompensatableSequenceActivity` activity. Add an exception handler that traps the `System.Exception` exception and contains a `CompensateActivity` activity with a `TargetPropertyName` property that points to the second `CompensatableSequenceActivity` activity.

**Correct Answers: A**