

Pass Microsoft 70-487 Exam

Number: 70-487
Passing Score: 800
Time Limit: 120 min
File Version: 42.0



Pass Microsoft 70-487 Exam

Exam Name: Developing Windows Azure and Web Services

General Exam

QUESTION 1

You are building an ADO.NET Entity Framework application. You need to validate the conceptual schema definition language (CSDL), store schema definition language (SSDL), and mapping specification language (MSL) files. Which Entity Data Model tool can you use? (Each correct answer presents a complete solution.

Choose all that apply.)

- A. EDM Generator (EdmGen.exe)
- B. ADO.NET Entity Data Model Designer
- C. Entity Data Model Wizard
- D. Update Model Wizard

Correct Answer: BC

Section: (none)

Explanation

Explanation/Reference:

QUESTION 2

DRAG DROP

You are developing an ASP.NET Web API action method.

The action method must return the following JSON in the message body.

```
{ " Name ":" Fabrikam", "Vendor Id": 9823, "Items": ["Apples", "Oranges"] }
```

You need to return an anonymous object that is serialized to JSON.

What should you do? (To answer, drag the appropriate code segments to the correct location or locations in the answer area. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

```

"Fabrikam", VendorNumber = 9823,
"Fabrikam", VendorNumber = "9823",
new List<string> { "Apples", "Oranges" }
new List<string> { "Apples, Oranges" }
return new List<string>
return new
        
```

Answer Area

```

public object Get()
{
    {
        Name = 
        Items = 
    };
}
        
```

A.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

```

public object Get()
{
    return new
    {
        Name = "Fabrikam", VendorNumber = 9823,
        Items = new List<string> { "Apples", "Oranges" }
    };
}
        
```

QUESTION 3

You are developing an ASP.NET MVC web application that contains the following HTML.

```
<table id= "customer" ></table>
```

You also have an ASP.NET Web API application that contains a call for retrieving customers. You must send and retrieve the data in the most compact format possible. You need to update the HTML for the customers table to contain data from the Web API application. Which script segment should you use?

A.

```
<script>
$(function () {
    var $customers = $("#customers");
    $.ajax({
        url: "api/customers",
        dataType: "json",
        success: function (data) {
            ...
        }
    });
});
</script>
```

B.

```
<script>
$(function () {
    var $customers = $("#customers");
    $.xml({
        url: "api/customers",
        dataType: "ajax",
        success: function (data) {
            ...
        }
    });
});
</script>
```

C.

```
<script>
$(function () {
    var $customers = $("#customers");
    $.json({
        url: "api/customers",
        dataType: "ajax",
        success: function (data) {
            ...
        }
    });
});
</script>
```

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

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 4

DRAG DROP

You are developing an ASP.NET MVC Web API application. The method names of the Web API must match naming guidelines for RESTful services. You need to create methods to support standard insert, select, update, and delete operations in an HTTP service. What should you do? (To answer, drag the appropriate HTTP methods to the correct row in the table in the answer area. Each HTTP method may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)

Answer Area		
Action	HTTP method	Relative URI
Retrieve a list of all customers	<input type="text"/>	/api/customers
Retrieve a customer by id	<input type="text"/>	/api/customers/ <i>id</i>
Retrieve a customer by category	<input type="text"/>	/api/customer/?category= <i>category</i>
Create a new customer	<input type="text"/>	/api/customers
Update a customer	<input type="text"/>	/api/customers/ <i>id</i>
Remove a customer	<input type="text"/>	/api/customers/ <i>id</i>

A.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

Put is missing

Action	HTTP method	Relative URI
Retrieve a list of all customers	GET	/api/customers
Retrieve a customer by id	GET	/api/customers/ <i>id</i>
Retrieve a customer by category	GET	/api/customer/?category= <i>category</i>
Create a new customer	POST	/api/customers
Update a customer	?? PUT ?	/api/customers/ <i>id</i>
Remove a customer	DELETE	/api/customers/ <i>id</i>

QUESTION 5

You develop an ASP.NET MVC application that is secured by using SSL. You are ready to deploy the application to production. The deployment package must include the installation of the SSL certificate. You need to configure the deployment package to meet the requirement. What should you do?

- A. Create a web publish pipeline target file with a custom web deploy target.
- B. In the Package/Publish settings of the project, select the All Files in this project option.
- C. Extend the CopyAllFilesToSingleFolder target in the project file.
- D. In the Build Events settings of the project, configure a pre-build event to include the SSL certificate.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 6

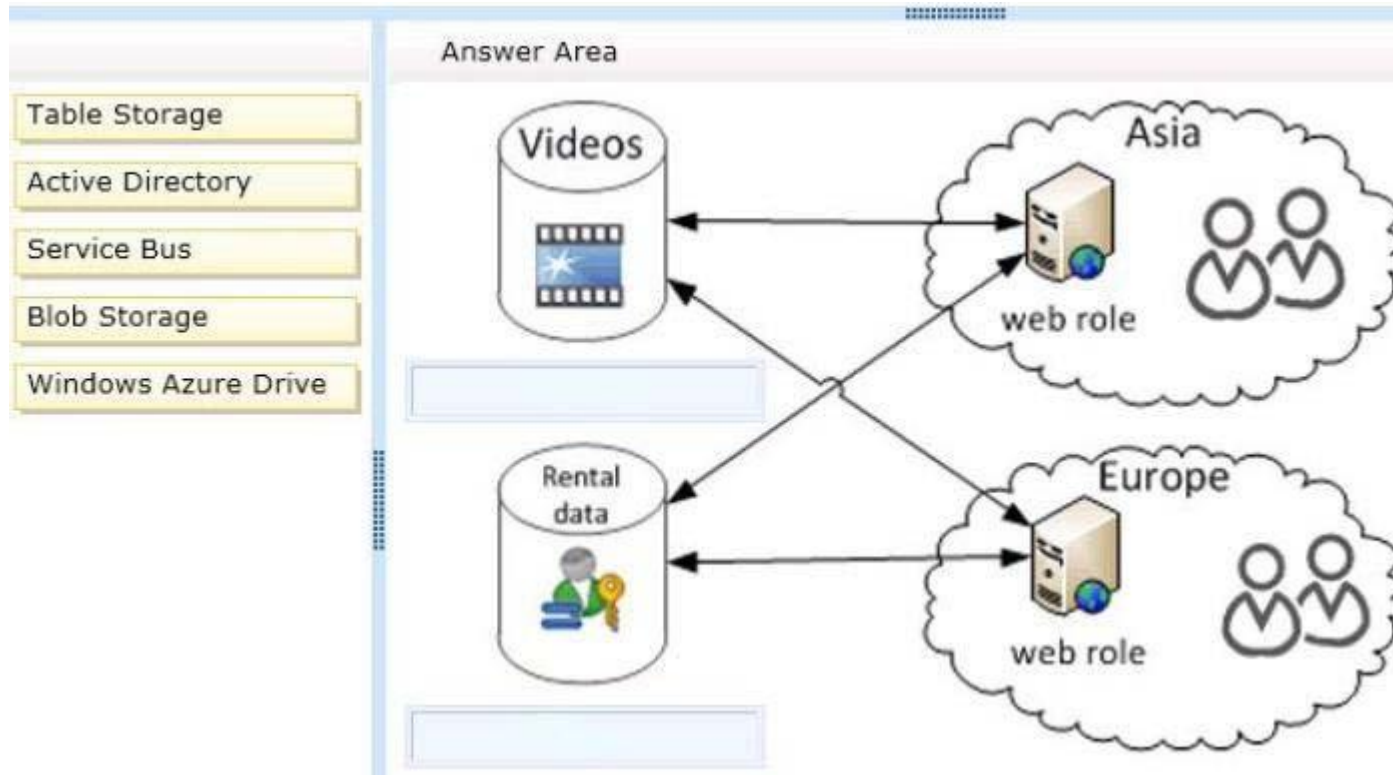
DRAG DROP

You are developing a Windows Azure based web application that provides users the ability to rent training videos. The application is deployed to hosted services in Asia and Europe.

The web application must meet the following requirements:

- Video files are large and must be able to be streamed.
- Streaming videos requires low latency network connections.
- Rental data contains structured information about the user and the video.
- Rental permissions are checked every five seconds during video playback.

You need to recommend storage architecture for the application. What should you do? (To answer, drag the appropriate technologies to the correct location or locations in the answer area. Each technology may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)



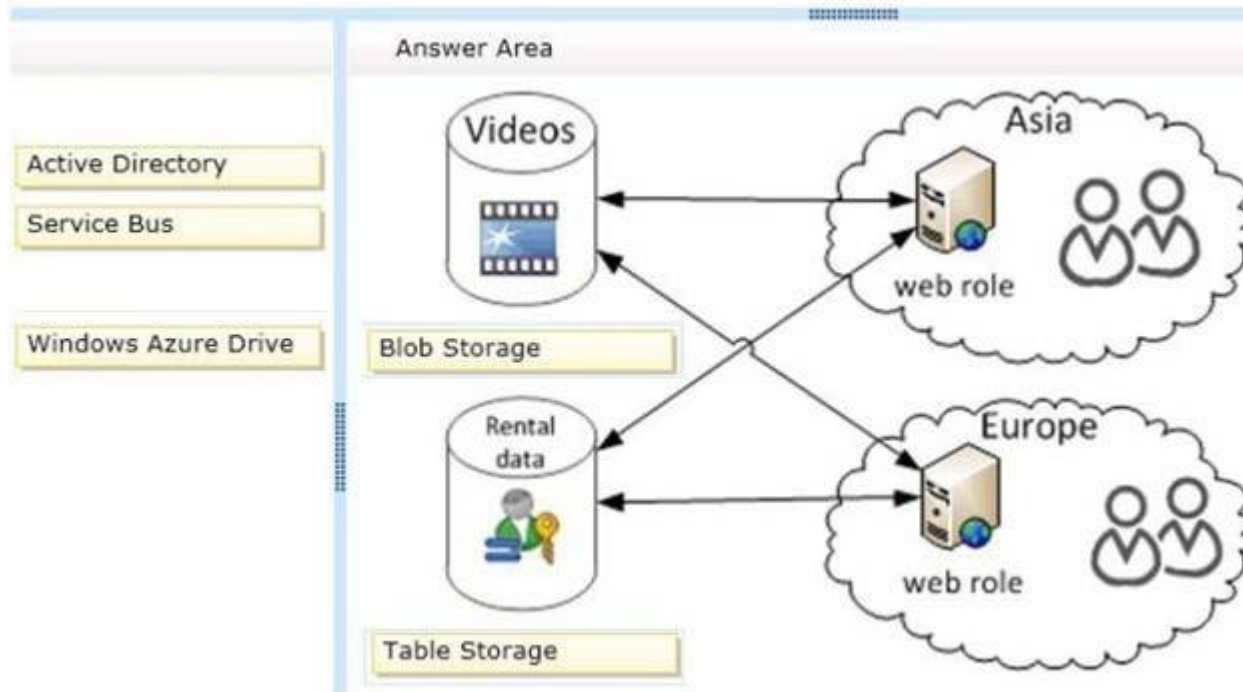
A.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

**QUESTION 7**

You are developing a WCF service. A new service instance must be created for each client request. You need to choose an instancing mode. Which instancing mode should you use?

- A. Single
- B. PerRequest
- C. PerCall
- D. Multiple
- E. PerSession

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 8

You are designing an ASP.NET Web API application. You need to select an HTTP verb to allow blog administrators to remove a comment. Which HTTP verb should you use?

- A. PUT
- B. DELETE
- C. POST
- D. GET

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 9

You are developing an ASP.NET MVC application. The application is an order processing system that uses the ADO.NET Entity Framework against a SQL Server database. It has a controller that loads a page that displays all orders along with customer information. Lazy loading has been disabled. The Order class is shown below.

```
public partial class Order
{
    ...
    public string CustomerID { get; set; }
    ...
    public virtual Customer Customer { get; set; }
}
```

You need to return the orders and customer information in a single round trip to the database. Which code segment should you use?

- ☐ A.

```
public ActionResult Index()
{
    IQueryable<Order> orders = db.Orders;
    orders = orders.Include("Customer");
    return View(orders.ToList());
}
```
- ☐ B.

```
public ActionResult Index()
{
    IQueryable<Order> orders = db.Orders.Include("Order.Customer");
    return View(orders.ToList());
}
```
- ☐ C.

```
public ActionResult Index()
{
    IQueryable<Order> orders = db.Orders;
    orders.Select(o => o.Customer).Load();
    return View(orders.ToList());
}
```
- ☐ D.

```
public ActionResult Index()
{
    IQueryable<Order> orders = db.Orders;
    return View(orders.ToList());
}
```

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

Correct Answer: A

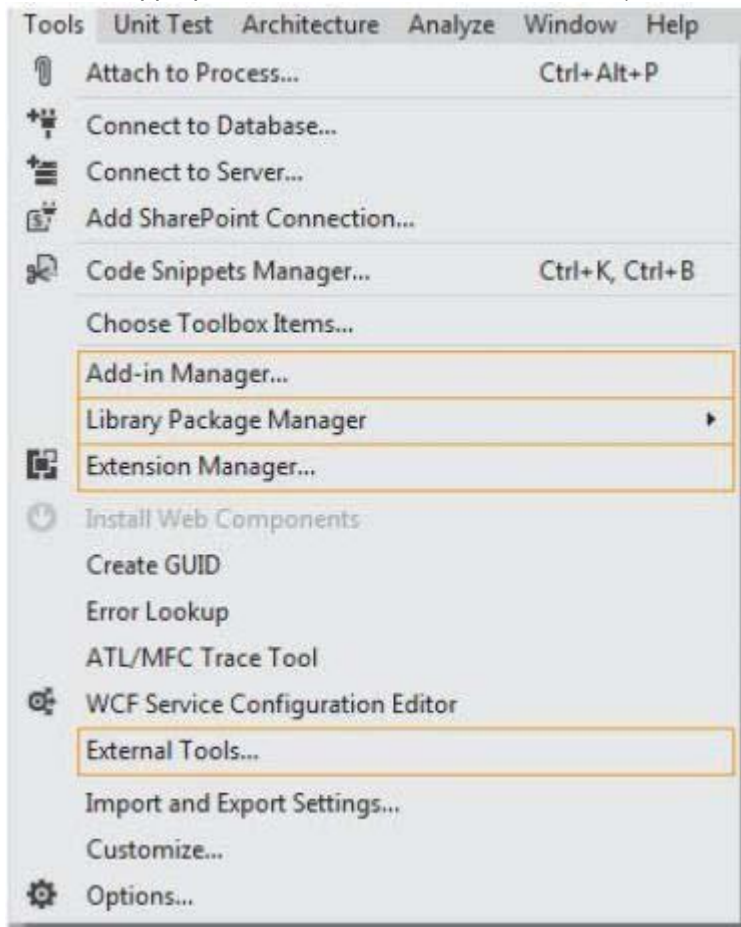
Section: (none)

Explanation

Explanation/Reference:

QUESTION 10
HOTSPOT

You are supporting an application that uses the ADO.NET Entity Framework to query and access data. The latest version of Entity Framework contains bug fixes that will improve performance. You need to update Entity Framework. Which Visual Studio 2012 menu item should you choose? (To answer, select the appropriate menu item in the answer area.)



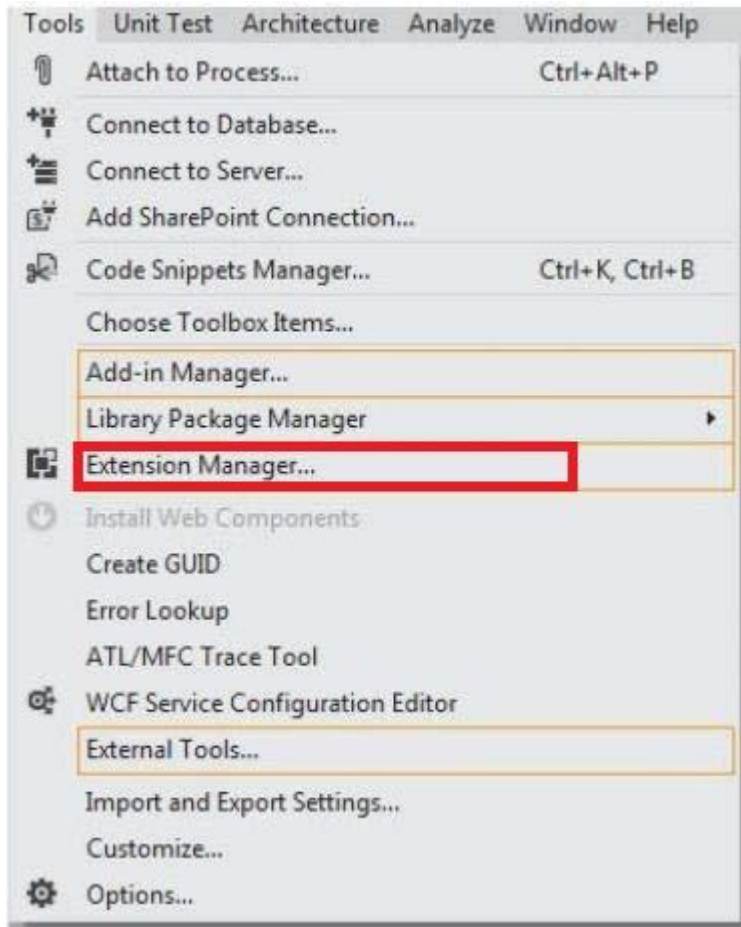
- A. Add-In Manager...
- B. Library Package Manager
- C. Extension Manager...
- D. External Tools...

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:



QUESTION 11

You are developing an ASP.NET MVC application. Deployment administrators do not have access to Visual Studio 2010, but will have the elevated permissions required to deploy the application to the servers. You need to select a deployment tool for use by the deployment administrators. Which tool should you use?

- A. Publish Web Site Tool
- B. Web Deployment Package
- C. One-Click Publish
- D. Deployment Package Editor

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

Case 1

QUESTION 1

Case Study: 1

Scenario 1

Background

You are developing a flight information consolidation service. The service retrieves flight information from a number of sources and combines them into a single data set. The consolidated flight information is stored in a SQL Server database. Customers can query and retrieve the data by using a REST API provided by the service. The service also offers access to historical flight information. The historical flight information can be filtered and queried in an ad hoc manner. The service runs on a Windows Azure Web Role. SSL is not used.

Business Requirements

- A new data source for historical flight information is being developed by a contractor located on another continent.
- If a time zone is not specified, then it should be interpreted as Coordinated Universal Time (UTC).
- When you upgrade a service from a staging deployment to a production deployment, the time that the service is unavailable must be minimized.
- The default port must be used for HTTP.

Technical Requirements

The existing sources of flight information and the mechanism of exchange are listed below.

- Blue Yonder Airlines provides flight information in an XML file.
- Consolidated Messenger provides flight information in a Microsoft Access database that is uploaded every 12 hours to the service using SFTP. The company uses port 22 for SFTP.
- Margie's Travel provides and consumes flight information using serialized ADO.NET DataSets. Data is periodically synced between the service and Margie's Travel.
- Trey Research provides data from multiple sources serialized in proprietary binary formats. The data must be read by using .NET assemblies provided by Trey Research. The assemblies use a common set of dependencies. The current version of the Trey Research assemblies is 1.2.0.0. All assemblies provided by Trey Research are signed with a key pair contained in a file named Trey.snk, which Trey Research also supplies.
- The application specification requires that any third-party assemblies must have strong names.

Application Structure

FlightInfo.cs

```
public class FlightInfo
{
    string DataSource { get; set; }
    public string Airline { get; set; }
    public string Flight { get; set; }
    public DateTimeOffset Arrival { get; set; }
    public int Seats { get; set; }
    public bool WasLate { get; set; }
}
```

BlueYonderLoader.cs

```
public class BlueYonderLoader
{
    public IEnumerable<RawFlightData> LoadFlights(XDocument feed)
    {
        ...
    }

    private RawFlightData Parse(XElement flightElement)
    {
        ...
    }
}
```

HistoricalDataLoader.cs

```
public class HistoricalDataLoader
{
    public static IEnumerable<HistoricalFlightInfo> LoadHistoricalFlights()
    {
        ...
    }

    public void StreamHistoricalFlights(XmlWriter responseWriter, string airline)
    {
        ...
    }

    private XElement ConvertToHistoricalFlight(XElement flight)
    {
        return new XElement("Flight", flight);
    }

    private string GetAirline(XElement flightName)
    {
        return flightName.Value.Substring(0, 2);
    }

    IEnumerable<XElement> RemoteDataStream()
    {
        return XDocument.Load("").Elements();
    }
}
```

MargiesTravelSync.cs

```
public class MargiesTravelSync
{
    public void Sync()
    {
        ...
    }

    private DataSet LoadLocal()
    {
        var dataSet = new DataSet();
        dataSet.ReadXml("local");
        return dataSet;
    }

    private StreamWriter SendStream()
    {
        return new StreamWriter("SendStream");
    }

    private StreamReader ReceiveStream()
    {
        return new StreamReader("ReceiveStream");
    }
}
```



FlightInfoContext.cs

```
public class FlightInfoContext : DbContext
{
    public DbSet<FlightInfo> FlightInfo { get; set; }

    public override int SaveChanges()
    {
        return base.SaveChanges();
    }

    private bool IsTransient(int ex)
    {
        var errors = new[] { 10053, 10054, 64 };
        return errors.Contains(ex);
    }
}
```

FlightDataController.cs

```
public class FlightDataController : ApiController
{
    FlightInfoContext _Context;

    public FlightDataController()
    {
        _Context = new FlightInfoContext();
    }

    [HttpGet]
    public IEnumerable<FlightInfo> GetFlightInfo()
    {
        return _Context.FlightInfo.Select(x => x).AsEnumerable();
    }

    private IEnumerable<HistoricalFlightInfo> LoadHistorical()
    {
        return HistoricalDataLoader.LoadHistoricalFlights();
    }
}
```

A.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 2

DRAG DROP

You need to configure the Windows Azure service definition to enable Consolidated Messenger to upload files. What should you do? (To answer, drag the appropriate configuration items to the correct location or locations. Each configuration item may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)

Answer Area

http

tcp

https

InternalEndpoint

InputEndpoint

80

22

3389

```

<Binding name="Website" endpointName="Website" />
<Binding name="Transfer" endpointName="Transfer" />
</Bindings>
</Site>
</Sites>
<Endpoints>

< <input type="text" value="" /> name="Website"

protocol=" <input type="text" value="" /> "

port=" <input type="text" value="" /> " />

< <input type="text" value="" /> name="Transfer"

protocol=" <input type="text" value="" /> "

port=" <input type="text" value="" /> " />

</Endpoints>
</WebRole>

```

A.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

"InputEndpoint", "http", "80"

"InputEndpoint", "tcp", "22"

Credits to Rem

QUESTION 3

DRAG DROP

Flight information data provided by Margie's Travel is updated both locally and remotely. When the data is synced, all changes need to be merged together without causing any data loss or corruption. You need to implement the Sync() method in the MargiesTravelSync.es file. What should you do? (To answer, drag the appropriate code segments to the correct location or locations in the answer area. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)

XmlReadMode.DiffGram

XmlReadMode.Fragment

XmlReadMode.InferSchema

XmlWriteMode.DiffGram

XmlWriteMode.IgnoreSchema

Answer Area

```
public void Sync()  
{  
    var sendStream = SendStream();  
    var receiveStream = ReceiveStream();  
    var local = LoadLocal();  
  
    local.WriteXml(sendStream, );  
  
    local.ReadXml(receiveStream, );  
}
```

A.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

XmlWriteMode.DiffGram and XmlReadMode.DiffGram

QUESTION 4

DRAG DROP

The service has been deployed to Windows Azure. Trey Research has provided version 1.3.0.0 of the assembly to support a change in the serialization format. The service must remain available during the transition to the new serialization format. You need to ensure that the service is using the new assembly. Which configuration setting should you add to the web.config? (To answer, drag the appropriate configuration elements to the correct location or locations in the answer area. Each configuration element may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)

```
codeBase version="1.3.0.0" href="Trey.Serialization.dll"
```

```
bindingRedirect oldVersion="1.2.5.0" newVersion="1.3.0.0"
```

```
bindingRedirect oldVersion="1.2.0.0" newVersion="1.3.0.0"
```

```
runtime
```

```
location
```

```
< >
```

```
<assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">  
  <dependentAssembly>  
    <assemblyIdentity name="Trey.Serialization" />  
  </dependentAssembly>  
</assemblyBinding>
```

```
<
```

```
</dependentAssembly>  
</assemblyBinding>
```

```
</ >
```

A.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

```
< runtime >

<assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
  <dependentAssembly>
    <assemblyIdentity name="Trey.Serialization" />
    < bindingRedirect oldVersion="1.2.0.0" newVersion="1.3.0.0"

  </dependentAssembly>
</assemblyBinding>

</ runtime >
```

QUESTION 5

You are adding a new REST service endpoint to the FlightDataController controller. It returns flights from the consolidated data sources only for flights that are late. You need to write a LINQ to Entities query to extract the required data. Which code segment should you use?

- ☐ A.

```
var historical = LoadHistorical();  
var query = _Context.FlightInfo.AsQueryable()  
    .Join(historical, x => x.Flight, y => y.Flight, (x, y) => new { Current = x,  
    Historical = y })  
    .Where(x => x.Historical.WasLate)  
    .Select(x => x.Current);
```
- ☐ B.

```
var historical = LoadHistorical();  
var query = _Context.FlightInfo.AsEnumerable()  
    .Where(x => historical.All(y => y.WasLate && x.Flight == y.Flight))  
    .Select(x => x);
```
- ☐ C.

```
var historical = LoadHistorical();  
var query = _Context.FlightInfo.AsQueryable()  
    .Where(x => historical.Select(y => y.Flight).Contains(x.Flight))  
    .Where(x => historical.Any(y => y.WasLate))  
    .Select(x => x);
```
- ☐ D.

```
var historical = LoadHistorical();  
var query = _Context.FlightInfo.AsEnumerable()  
    .Join(historical, x => x.Flight, y => y.Flight, (x, y) => new { Current = x,  
    Historical = y })  
    .Where(x => x.Historical.WasLate)  
    .Select(x => x.Current);
```

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

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

D is right because you send result as REST so if you use "AsQueryable" the result is deferred to the next enumeration of your result.

D is not optimized but will work.

A will break at runtime.

Credits to Rem

QUESTION 6

You need to load flight information provided by Consolidated Messenger. Which should you use?

- A. SQL Server Data Transformation Services (DTS)
- B. EntityTransaction and EntityCommand
- C. Office Open XML
- D. OleDbConnection and OleDbDataReader

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 7

You are adding a new REST service endpoint to the FlightDataController controller that returns the total number of seats for each airline. You need to write a LINQ to Entities query to extract the required data.

Which code segment should you use?

- ☐ A.

```
var query = from flight in _Context.FlightInfo
group flight by flight.Seats into agg
let airline = agg.First()
select new
{
    TotalSeats = agg.Key,
    Airline = airline,
};
```
- ☐ B.

```
var query = from flight1 in _Context.FlightInfo
from flight2 in _Context.FlightInfo
where flight1.Airline == flight2.Airline
select new
{
    Airline = flight1.Airline,
    TotalSeats = Math.BigMul(flight1.Seats, flight2.Seats),
};
```
- ☐ C.

```
var query = from flight in _Context.FlightInfo
from airline in flight.Airline
group airline by airline into agg
select new
{
    Airline = agg.Key,
    TotalSeats = agg.Sum(x => Convert.ToInt32(x)),
};
```
- ☐ D.

```
var query = from flight in _Context.FlightInfo
group flight by flight.Airline into agg
select new
{
    Airline = agg.Key,
    TotalSeats = agg.Sum(x => x.Seats),
};
```

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

D. Option D

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 8

You need to load flight information provided by Consolidated Messenger. What should you use?

- A. Office Open XML
- B. COM interop
- C. OleDbConnection and OleDbDataReader
- D. EntityConnection and EntityDataReader

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 9

Historical flight information data will be stored in Windows Azure Table Storage using the FlightInfo class as the table entity. There are millions of entries in the table. Queries for historical flight information specify a set of airlines to search and whether the query should return only late flights. Results should be ordered by flight name. You need to specify which properties of the FlightInfo class should be used at the partition and row keys to ensure that query results are returned as quickly as possible. What should you do? (Each correct answer presents part of the solution. Choose all that apply.)

- A. Use the WasLate property as the row key.
- B. Use the Airline property as the row key.
- C. Use the WasLate property as the partition key
- D. Use the Arrival property as the row key.
- E. Use the Airline property as the partition key.
- F. Use the Flight property as the row key.

Correct Answer: CF

Section: (none)

Explanation

Explanation/Reference:

QUESTION 10

Transformed historical flight information provided by the `RemoteDataStream()` method must be written to the response stream as a series of XML elements named `Flight` within a root element named `Flights`. Each `Flight` element has a child element named `FlightName` that contains the flight name that starts with the two-letter airline prefix. You need to implement the `StreamHistoricalFlights()` method so that it minimizes the amount of memory allocated. Which code segment should you use as the body of the `StreamHistoricalFlights()` method in the `HistoricalDataLoader.cs` file?

- ☐ A.

```
responseWriter.WriteStartElement("Flights");
var flights = RemoteDataStream()
    .OrderBy(x => GetAirline(x.Element("FlightName")));
var filteredFlights = flights
    .SkipWhile(x => GetAirline(x.Element("FlightName")) != airline);
foreach (var f in filteredFlights)
{
    var flight = ConvertToHistoricalFlight(f);
    flight.WriteTo(responseWriter);
}
responseWriter.WriteEndElement();
```
- ☐ B.

```
responseWriter.WriteStartElement("Flights");
var flights = RemoteDataStream().Select(x =>
{
    if (GetAirline(x) == airline)
    {
        return ConvertToHistoricalFlight(x);
    }
    return null;
});
flights.TakeWhile(x =>
{
    x.WriteTo(responseWriter);
    return x != null;
});
responseWriter.WriteEndElement();
```
- ☐ C.

```
var data = RemoteDataStream().ToDictionary(x =>
    GetAirline(x.Element("FlightName")),
    x => new XElement("Flights", ConvertToHistoricalFlight(x).Descendants()));
data[airline].WriteTo(responseWriter);
```
- ☐ D.

```
var flights = new XElement("Flights",
    from flight in RemoteDataStream()
    where GetAirline(flight.Element("FlightName")) == airline
    select ConvertToHistoricalFlight(flight));
flights.WriteTo(responseWriter);
```

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

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 11

Errors occasionally occur when saving data using the FlightInfoContext ADO.NET Entity Framework context. Updates to the data are being lost when an error occurs. You need to ensure that data is still saved when an error occurs by retrying the operation. No more than five retries should be performed. With which code segment should you replace the body of the SaveChanges() method in the FlightInfoContext.cs file?

- ☐ A.

```
var result = FlightInfo.SqlQuery("UPDATE WITH RETRY", FlightInfo, "IsTransient", 5);
if (result.Count() > 5)
{
    result.AsNoTracking();
    return -1;
}
return 0;
```
- ☐ B.

```
try
{
    return base.SaveChanges();
}
catch (EntityCommandExecutionException ex)
{
    if (ex.Data.Keys.Cast<int>().Any(x => IsTransient(x)))
    {
        return 5 & SaveChanges();
    }
    return -1;
}
```
- ☐ C.

```
for (var i = 0; i < 5; i++)
{
    try
    {
        return base.SaveChanges();
    }
    catch (SqlException ex)
    {
        if (IsTransient(ex.Number))
        {
            continue;
        }
    }
}
return base.SaveChanges();
```
- ☐ D.

```
var exception = new EntitySqlException();
while (exception.HResult != 0 && exception.Data.Count < 5)
{
    try
    {
        return base.SaveChanges();
    }
    catch (EntitySqlException ex)
```

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

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

EntitySQLException : Represents errors that occur when parsing Entity SQL command text. This exception is thrown when syntactic or semantic rules are violated.

SQLException : The exception that is thrown when SQL Server returns a warning or error. This class cannot be inherited.

EntityCommandExecutionException : Represents errors that occur when the underlying storage provider could not execute the specified command. This exception usually wraps a provider-specific exception.

Case 2

QUESTION 1

Case Study: 2

Scenario 2

Background

You are developing an ASP.NET MVC application in Visual Studio 2012 that will be used to process orders.

Business Requirements

The application contains the following three pages.

- A page that queries an external database for orders that are ready to be processed. The user can then process the order.
- A page to view processed orders.
- A page to view vendor information.

The application consumes three WCF services to retrieve external data.

Technical Requirements

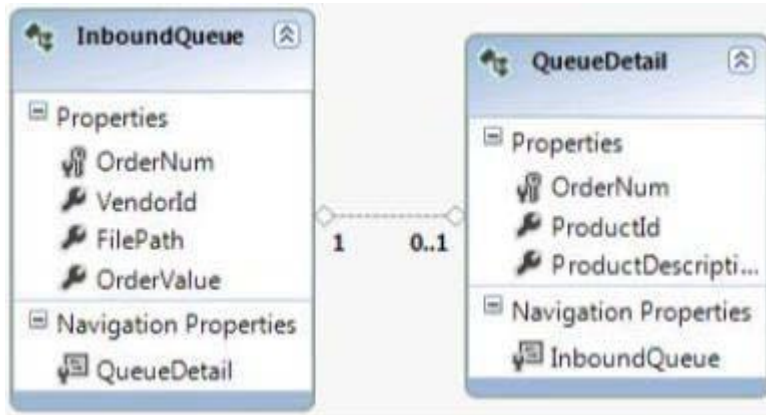
Visual Studio Solution:

The solution contains the following four projects.

- ExternalQueue: A WCF service project used to communicate with the external order database.
- OrderProcessor: An ASP.NET MVC project used for order processing and logging order metadata.
- OrderUpload: A WCF service project used to submit order data to an external data source.
- Shipping: A WCF service project used to acquire shipping information.

ExternalQueue Project:

Entity Framework is used for data access. The entities are defined in the ExternalOrders.edmx file as shown in the following diagram.



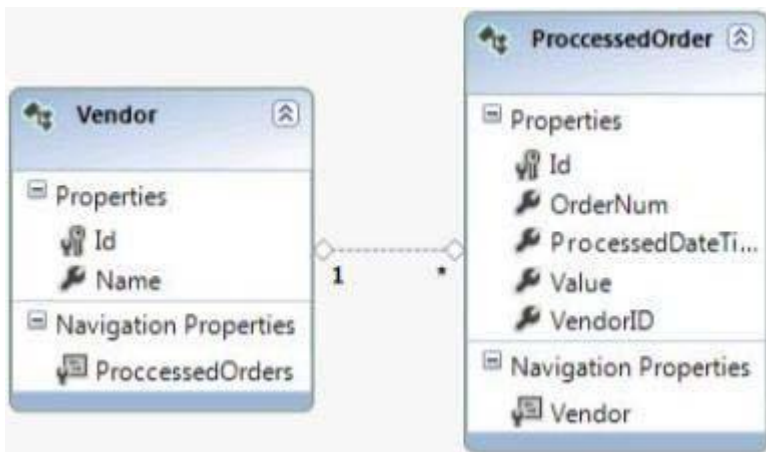
The project contains two services defined in the following files.

- IExternalQueueService.cs
- ExternalQueueService.svc.

The ExternalQueue.Helpers namespace contains a definition for a class named OrderNotFound Exception.

OrderProcessor Project:

Entity Framework is used for data access. The entities are defined in the ProcessedOrders.edmx file as shown in the following diagram.



The classes are contained in the OrderProcessor.Entities namespace. The project contains the following two controllers.

- InboundQueueController.cs
- ProcessedOrderController.cs

WCF service proxies to the ExternalQueue, Shipping and OrderUpload services have been generated by using the command prompt. The ExecuteCommandProcedure() method in the ExternalQueueService.svc file must run asynchronously.

The ProcessedOrderController controller has the following requirements.

The GetVendorPolicy() method must enforce a 10 minute absolute cache expiration policy.

The GetProcessedOrders() method must return a view of the 10 most recently processed orders.

OrderUpload Project:

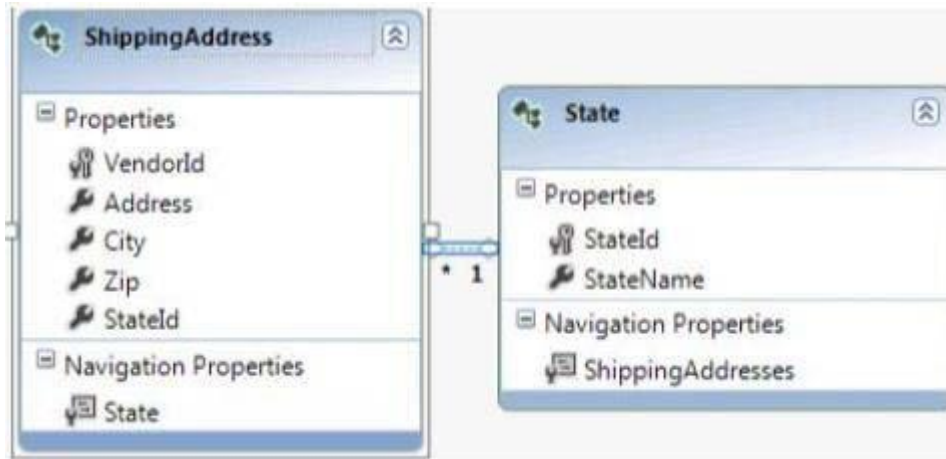
The project contains two services defined in the following files.

- IUploadCallbackService.cs
- UploadCallbackService.svc

Data Access is maintained in a file named UploadOrder.es.

Shipping Project:

Entity Framework is used for data access. The entities are defined in the ExternalOrders.edmx file as shown in the following diagram.



The Custom Tool property for ExternalOrders.edmx has been removed. POCO classes for the Entity Model are located in the ShippingAddress.cs file. The POCO entity must be loaded by using lazy loading. The project contains two services defined in the following files.

- IShippingService.cs
- ShippingService.svc.

The IShippingService contract must contain an operation that receives an order number as a parameter. The operation must return a class named ShippingInfo that inherits from a class named State.

Application Structure

ExternalQueue\IExternalQueueService.cs

```
IQ01 using System.Collections.Generic;
IQ02 using System.ServiceModel;
IQ03 using ExternalQueue.Helpers;
IQ04
IQ05 namespace ExternalQueue
IQ06 {
IQ07     [ServiceContract]
IQ08     public interface IExternalQueueService
IQ09     {
IQ10         [OperationContract]
IQ11         List<Entities.InboundQueue> GetExternalOrders();
IQ12
IQ13         [FaultContract(typeof(OrderNotFoundException))]
IQ14         [OperationContract]
IQ15         void DeleteExternalOrder(int orderNum);
IQ16
IQ17         [OperationContract]
IQ18         Entities.InboundQueue GetExternalOrder(int orderNum);
IQ19     }
IQ20 }
```

OrderProcessor\IExternalQueueService.svc

```
EQ01 using System;
EQ02 using System.Collections.Generic;
EQ03 using System.Linq;
EQ04 using System.Data.EntityClient;
EQ05 using System.Data;
EQ06 using ExternalQueue.Entities;
EQ07 using System.Data.Objects;
EQ08 using ExternalQueue.Helpers;
EQ09 using System.ServiceModel;
EQ10 using System.Threading.Tasks;
EQ11
EQ12 namespace ExternalQueue
EQ13 {
EQ14     public class ExternalQueueService : IExternalQueueService
EQ15     {
EQ16         public List<Entities.InboundQueue> GetExternalOrders()
EQ17         {
EQ18             List<InboundQueue> queueItems = new List<InboundQueue>();
EQ19             return queueItems;
EQ20         }
EQ21
EQ22         public void DeleteExternalOrder(int orderNum)
EQ23         {
EQ24             using (var context = new ExternalOrdersEntities())
EQ25             {
EQ26                 var orders = context.InboundQueues.Where(i => i.OrderNum ==
EQ27                 orderNum).ToList();
EQ28                 if (orders.Count() > 0)
EQ29                 {
EQ30                     using (EntityCommand cmd = new EntityCommand())
EQ31                     {
EQ32                         cmd.CommandText = "ExternalOrdersEntities.uspInboundQueueDelete";
EQ33                         cmd.CommandType = CommandType.StoredProcedure;
EQ34                         EntityParameter param = new EntityParameter();
EQ35                         param.Value = orderNum;
EQ36                         param.ParameterName = "orderNum";
EQ37                         cmd.Parameters.Add(param);
EQ38                         ExecuteCommandProcedure(cmd);
EQ39                     }
EQ40                 }
EQ41                 else
EQ42                 {
EQ43                     OrderNotFoundException ex = new OrderNotFoundException();
EQ44                     ex.OrderNum = orderNum;
EQ45                     ex.ExceptionMessage = "Order not found...Cannot delete";
EQ46                 }
EQ47             }
```

ExternalQueue\ProcessedOrderController.cs

```
PC01 using System;
PC02 using System.Collections.Generic;
PC03 using System.Linq;
PC04 using System.Runtime.Caching;
PC05 using System.Web.Mvc;
PC06 using OrderProcessor.Entities;
PC07 using OrderProcessor.Helpers;
PC08 using System.Configuration;
PC09
PC10 namespace OrderProcessor.Controllers
PC11 {
PC12     public class ProcessedOrderController : Controller
PC13     {
PC14         public ActionResult GetProcessedOrders()
PC15         {
PC16             using (var context = new ProcessedOrders())
PC17             {
PC18                 List<Entities.ProcessedOrder> orders = new List<ProcessedOrder>();
PC19                 return View(orders);
PC20             }
PC21         }
PC22
PC23         private ObjectCache cache {get { return MemoryCache.Default; }}
PC24
PC25         public ActionResult GetVendors()
PC26         {
PC27             List<Entities.Vendor> vendors = cache.Get
PC28             ("vendorKey") as List<Entities.Vendor>;
PC29             if (vendors == null)
PC30             {
PC31                 using (var context = new ProcessedOrders())
PC32                 {
PC33                     vendors = context.Vendors.ToList();
PC34                 }
PC35             }
PC36             return View(vendors);
PC37         }
PC38
PC39         private CacheItemPolicy GetVendorPolicy()
PC40         {
PC41             CacheItemPolicy vendorPolicy = new CacheItemPolicy();
PC42             return vendorPolicy;
```

OrderProcessor\InboundQueueController.cs

```
IC01 using System;
IC02 using System.Collections.Generic;
IC03 using System.Web.Mvc;
IC04 using OrderProcessor.Entities;
IC05 using ExternalQueue.Entities;
IC06 using System.ServiceModel;
IC07 using System.Collections;
IC08 using ExternalQueue.Helpers;
IC09 using OrderProcessor.Helpers;
IC10 using System.Linq;
IC11
IC12 namespace OrderProcessor.Controllers
IC13 {
IC14     public class InboundQueueController : Controller
IC15     {
IC16         public ActionResult GetQueueItems()
IC17         {
IC18             IEnumerable<InboundQueue> inboundOrders = Enumerable.Empty<InboundQueue>();
IC19             return View(inboundOrders);
IC20         }
IC21
IC22         public ActionResult ProcessOrder(int orderNum)
IC23         {
IC24             ExternalQueueServiceClient qService = new ExternalQueueServiceClient();
IC25             InboundQueue externalOrder = qService.GetExternalOrder(orderNum);
IC26             if (externalOrder != null)
IC27             {
IC28                 using (var context = new ProcessedOrders())
IC29                 {
IC30                     ProcessedOrder order = new ProcessedOrder();
IC31                     order.OrderNum = externalOrder.OrderNum;
IC32                     order.Value = Convert.ToDouble(externalOrder.OrderValue);
IC33                     order.VendorID = Convert.ToInt32(externalOrder.VendorId);
IC34                     order.ProcessedDateTime = DateTime.Now;
IC35                     context.ProcessedOrders.Add(order);
IC36                     context.SaveChanges();
IC37                 }
IC38                 qService.DeleteExternalOrder(orderNum);
IC39             }
IC40             return RedirectToAction("GetQueueItems");
IC41         }
IC42
IC43         public ActionResult ViewShippingInfo(int orderNum)
IC44         {
IC45             ShippingServiceClient shipService = new ShippingServiceClient();
IC46             var info = shipService.GetShippingInfo(orderNum);
```


OrderUpload\IUploadCallbackService.cs

```
IU01 using System.ServiceModel;
IU02
IU03 namespace OrderUpload
IU04 {
IU05     [ServiceContract(CallbackContract = typeof(IUploadCallback))]
IU06     public interface IUploadCallbackService
IU07     {
IU08         [OperationContract]
IU09         void UploadOrder(int orderNum);
IU10     }
IU11
IU12     public interface IUploadCallback
IU13     {
IU14         [OperationContract]
IU15         decimal GetOrderValue(int orderNum);
IU16     }
IU17 }
```

OrderUpload\UploadCallbackService.svc

```
US01 using System.ServiceModel;
US02
US03 namespace OrderUpload
US04 {
US05     public class UploadCallbackService : IUploadCallbackService
US06     {
US07         public void UploadOrder(int orderNum)
US08         {
US09         }
US10     }
US11 }
```

Shipping\IShippingService.cs

```
IS01 using System.Runtime.Serialization;
IS02 using System.ServiceModel;
IS03
IS04 namespace Shipping
IS05 {
IS06     public interface IShippingService
IS07     {
IS08     }
IS09 }
IS10 }
```

Shipping\ShippingAddress.cs

```
SA01 using System.Collections.Generic;
SA02 using System.Data.Objects;
SA03
SA04 namespace Shipping.POCO
SA05 {
SA06     public class ShippingAddress
SA07     {
SA08         public int VendorId { get; set; }
SA09         public string Address { get; set; }
SA10         public string City { get; set; }
SA11         public int StateId { get; set; }
SA12         public string Zip { get; set; }
SA13         public State State { get; set; }
SA14     }
SA15
SA16     public class State
SA17     {
SA18         public int StateId { get; set; }
SA19         public string StateName { get; set; }
SA20         public List<ShippingAddress> ShippingAddresses { get; set; }
SA21     }
SA22 }
```

A.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 2

DRAG DROP

The GetVendorPolicy() private method in the ProcessedOrderController controller is returning a CacheItemPolicy object with default values. The returned policy must expire if the external file located at C:\Triggers\VendorTrigger.txt has been modified or the timeout outlined in the technical requirements is reached. You need to return the policy. How should you build the method? (To answer, drag the appropriate code segments to the correct location or locations in the answer area. Each code

segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)

Priority

ChangeMonitors

AbsoluteExpiration

Expiration

DateTime.AddMinutes

DateTime.Now.AddMinutes

Answer Area

```
private CacheItemPolicy GetVendorPolicy()
{
    CacheItemPolicy vendorPolicy = new CacheItemPolicy();

    vendorPolicy.

    =  (10);

    vendorPolicy.

    .Add(new HostFileChangeMonitor(GetTriggerPaths()));

    return vendorPolicy;
}
```

A.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:


```
private CacheItemPolicy GetVendorPolicy()
{
    CacheItemPolicy vendorPolicy = new CacheItemPolicy();

    vendorPolicy.AbsoluteExpiration
        = DateTime.Now.AddMinutes(10);

    vendorPolicy.ChangeMonitors
        .Add(new HostFileChangeMonitor(GetTriggerPaths()));

    return vendorPolicy;
}
```

QUESTION 3

The GetExternalOrder() method in the ExternalQueueService service is throwing a runtime error. The method must query the database for a record that matches the orderNum parameter passed to the method.

You need to modify the queryString string to retrieve the record. With which code segment should you replace line EQ64?

- ☐ A.

```
string queryString = @"SELECT VALUE q FROM ExternalOrdersEntities.InboundQueues AS q
WHERE q.OrderNum = @orderNum";
```
- ☐ B.

```
string queryString = @"SELECT VALUE * FROM ExternalOrdersEntities.InboundQueues
WHERE OrderNum = @orderNum";
```
- ☐ C.

```
string queryString = @"SELECT q.OrderNum, q.VendorId, q.FilePath, q.OrderValue
FROM ExternalOrdersEntities AS q WHERE q.OrderNum = @orderNum";
```
- ☐ D.

```
string queryString = @"SELECT q FROM ExternalOrdersEntities.InboundQueues
WHERE q.OrderNum = @orderNum";
```

A. Option A

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

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 4

DRAG DROP

You add a class named ShippingInfo. You need to modify the IShippingService interface and the ShippingInfo class to meet the technical requirements. What should you do? (To answer, drag the appropriate code segments to the correct location or locations in the answer area. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)



[DataMember]

[CollectionDataContract]

[DataContract]

[ServiceContract]

[OperationContract]

Answer Area

```
public interface IShippingService
{
    ShippingInfo GetShippingInfo(int orderNum);
}

public class State
{
    public string StateName { get; set; }
}

public class ShippingInfo : State
{
    public string StreetAddress { get; set; }

    public string ZipCode { get; set; }
}
```

A.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

```
[ServiceContract]
public interface IShippingService
{
    [OperationContract]
    ShippingInfo GetShippingInfo(int orderNum);
}

[DataContract]
public class State
{
    [DataMember]
    public string StateName { get; set; }
}

[DataContract]
public class ShippingInfo : State
{
    [DataMember]
    public string StreetAddress { get; set; }

    [DataMember]
    public string ZipCode { get; set; }
}
```

QUESTION 5

The DeleteExternalOrder() method in the ExternalQueueService service is not throwing a FaultException exception as defined by the FaultContractAttribute attribute in the IExternalQueueService.cs file. You need to throw the FaultException exception. Which code segments can you insert at line EQ45 to achieve this goal? (Each correct answer presents a complete solution. Chose all that apply)

- ☐ A. `throw new FaultException<OrderNotFoundException>(ex.ExceptionMessage);`
- ☐ B. `throw new FaultException<OrderNotFoundException>(ex, new FaultReason("Order not found."));`
- ☐ C. `throw new FaultException<OrderNotFoundException>(ex);`
- ☐ D. `throw new FaultException
(new OrderNotFoundException(new Exception(ex.ExceptionMessage)), "Order not
found.");`

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

Correct Answer: BC

Section: (none)

Explanation

Explanation/Reference:

<http://msdn.microsoft.com/en-us/library/ms586019.aspx>

QUESTION 6

The DeleteExternalOrder() method in the ExternalQueueService service is not throwing a FaultException exception as defined by the FaultContractAttribute attribute in the IExternatQueueService.cs file. You need to throw the FaultException exception. Which code segment can you insert at line EQ45 to achieve this goal? (Each correct answer presents a complete solution. Chose all that apply.)

- ☐ A. `string queryString = @"SELECT q.OrderNum, q.VendorId, q.FilePath, q.OrderValue
FROM ExternalOrdersEntities.InboundQueues AS q WHERE q.OrderNum = @orderNum";`
- ☐ B. `string queryString = @"SELECT * FROM ExternalOrdersEntities.InboundQueues
WHERE OrderNum = @orderNum";`
- ☐ C. `string queryString = @"SELECT VALUE q FROM ExternalOrdersEntities.InboundQueues AS q
WHERE q.OrderNum = @orderNum";`
- ☐ D. `string queryString = @"SELECT VALUE FROM ExternalOrdersEntities.InboundQueues
WHERE OrderNum = @orderNum";`

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

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 7

The QueueDetail entity type must inherit from the InboundQueue entity type in the ExternalQueue service project using table-per-type inheritance. You need to modify the entities in the designer. What should you do? (Each correct answer presents part of the solution. Choose all that apply.)

- A. Remove the OrderNum property in InboundQueue.
B. Remove the OrderNum property in QueueDetail.
C. Set the QueueDetail BaseType to InboundQueue.
D. Remove the association between the entities.
E. Right-click the entities and validate the table mapping.
F. Set the InboundQueue BaseType to QueueDetail.

Correct Answer: BCDE

Section: (none)

Explanation

Explanation/Reference:

QUESTION 8

DRAG DROP

The UploadOrder() method in the UploadCallbackService service is not implementing the callback behavior defined in the IUploadCallBackService interface. You need to modify the class to implement the required callback behavior. What should you do? (To answer, drag the appropriate code segments to the



correct location or locations in the answer area. Each code segments may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)

Multiple

Single

GetOrderValue

UploadCallbackService

IUploadCallback

Answer Area

```
[ServiceBehavior(ConcurrencyMode =
    ConcurrencyMode. )]

public class UploadCallbackService : IUploadCallbackService
{
    public void UploadOrder(int orderNum)
    {
        callback = OperationContext
            .Current.GetCallbackChannel< >();
        decimal value = callback. orderNum);

        UploadDB.UploadOrder.Upload(orderNum, value);
    }
}
```

A.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:


```
[ServiceBehavior(ConcurrencyMode =  
    ConcurrencyMode.Single)]  
  
public class UploadCallbackService : IUploadCallbackService  
{  
    public void UploadOrder(int orderNum)  
    {  
        IUploadCallback callback = OperationContext  
            .Current.GetCallbackChannel<IUploadCallback>();  
        decimal value = callback.GetOrderValue(orderNum);  
  
        UploadDB.UploadOrder.Upload(orderNum, value);  
    }  
}
```