

MB7-701

Number: MB7-701
Passing Score: 700
Time Limit: 995 min
File Version: 25.4



<http://www.gratisexam.com/>



Microsoft MB7-701 Questions & Answers

Exam Name: Microsoft Dynamics NAV 2013 Core Setup and Finance

Passguide

QUESTION 1

You are creating a Windows Communication Foundation (WCF) service that is implemented as follows. (Line numbers are included for reference only.)

```
01 [ServiceContract]
02 [ServiceBehavior(IncludeExceptionDetailsInFaults = true)]
03 public class OrderService
04 {
05     [OperationContract]
06     public void SubmitOrder(Order anOrder)
07     {
08         try
09         {
10             ...
11         }
12         catch(DivideByZeroException ex)
13         {
14             ...
15         }
16     }
17 }
```

You need to ensure that the stack trace details of the exception are not included in the error information sent to the client.

What should you do?

- A. Replace line 14 with the following line:
`throw;`
- B. Replace line 14 with the following line:
`throw new FaultException<Order>(anOrder, ex.ToString());`
- C. After line 05, add the following line:
`[FaultContract(typeof(FaultException<Order>))]`

Replace line 14 with the following line:
`throw ex;`
- D. Alter line 05, add the following line:
`[FaultContract(typeof(FaultException<Order>))]`

Replace line 14 with the following line:
`throw new FaultException<Order>(anOrder, "Divide by zero exception");`

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

Typical deployed services use the `FaultContractAttribute` to formally specify all SOAP faults that a client can expect to receive in the normal course of an operation.

Error information in a `FaultContractAttribute` appears as a `FaultException<TDetail>` (where the typeparameter is the serializable error object specified in the operation's `FaultContractAttribute`) when it arrives at a client application.

The `FaultContractAttribute` can be used to specify SOAP faults for both two-way service methods and for asynchronous method pairs.

Because `FaultException<TDetail>` is both a `FaultException` and therefore a `CommunicationException`, to catch specified SOAP faults make sure you catch the `FaultException<TDetail>` types prior to the `FaultException` and `CommunicationException` types or handle the specified exceptions in one of those exception handlers.

FaultException<TDetail> Class

(<http://msdn.microsoft.com/en-us/library/ms576199.aspx>)

QUESTION 2

You are creating a Windows Communication Foundation (WCF) service. You do not want to expose the internal implementation at the service layer.

You need to expose the following class as a service named Arithmetic with an operation named Sum:

```
public class Calculator
{
    public int Add(int x, int y)
    {
    }
}
```

Which code segment should you use?

- A. `[ServiceContract(Namespace="Arithmetic")]`
public class Calculator
{
 [OperationContract(Action="Sum")]
 public int Add(int x, int y)
 {}
}
- B. `[ServiceContract(ConfigurationName="Arithmetic")]`
public class Calculator
{
 [OperationContract(Action="Sum")]
 public int Add(int x, int y)
 {}
}
- C. `[ServiceContract(Name="Arithmetic")]`
public class Calculator
{
 [OperationContract(Name="Sum")]
 public int Add(int x, int y)
 {}
}
- D. `[ServiceContract(Name="Arithmetic")]`
public class Calculator
{
 [OperationContract(ReplyAction="Sum")]
 public int Add(int x, int y)
 {}
}

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

Name Gets or sets the name for the <portType> element in Web Services Description Language (WSDL). (<- ServiceContract)

Name Gets or sets the name of the operation. (<-OperationContract)

ServiceContractAttribute Class

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.servicecontractattribute.aspx>)

OperationContractAttribute Class

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.operationcontractattribute.aspx>)

QUESTION 3

You are developing a data contract for a Windows Communication Foundation (WCF) service. The data in the data contract must participate in round trips. Strict schema validity is not required. You need to ensure that the contract is forward-compatible and allows new data members to be added to it.

Which interface should you implement in the data contract class?

- A. `ICommunicationObject`
- B. `IExtension<T>`
- C. `IExtensibleObject<T>`
- D. `IExtensibleDataObject`

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

IExtensibleDataObject Interface Provides a data structure to store extra data encountered by the `XmlObjectSerializer` during deserialization of a type marked with the `DataContractAttribute` attribute.

The `IExtensibleDataObject` interface provides a single property that sets or returns a structure used to store data that is external to a data contract. The extra data is stored in an instance of the `ExtensionDataObject` class and accessed through the `ExtensionData` property. In a roundtrip operation where data is received, processed, and sent back, the extra data is sent back to the original sender intact. This is useful to store data received from future versions of the contract. If you do not implement the interface, any extra data is ignored and discarded during a roundtrip operation.

IExtensibleDataObject Interface

(<http://msdn.microsoft.com/en-us/library/system.runtime.serialization.iextensibledataobject.aspx>)

```
// Implement the IExtensibleDataObject interface to store the extra data for future versions.
```

```
[DataContract(Name = "Person", Namespace = "http://www.cohowinery.com/employees")]
```

```
class Person : IExtensibleDataObject
```

```
{
```

```
    // To implement the IExtensibleDataObject interface,  
    // you must implement the ExtensionData property. The property  
    // holds data from future versions of the class for backward compatibility.
```

```
    private ExtensionDataObject extensionDataObject_value;
```

```
    public ExtensionDataObject ExtensionData
```

```
    {
```

```
        get
```

```
        {
```

```
            return extensionDataObject_value;
```

```
        }
```

```
        set
```

```
        {
```

```
            extensionDataObject_value = value;
```

```
        }
```

```
    }
```

```
    [DataMember]
```

```
    public string Name;
```

```
}
```

Forward-Compatible Data Contracts

(<http://msdn.microsoft.com/en-us/library/ms731083.aspx>)

QUESTION 4

A Windows Communication Foundation (WCF) application uses a data contract that has several data

members.

You need the application to throw a `SerializationException` if any of the data members are not present when a serialized instance of the data contract is deserialized.

What should you do?

- A. Add the `KnownType` attribute to the data contract.
Set a default value in each of the data member declarations.
- B. Add the `KnownType` attribute to the data contract.
Set the `Order` property of each data member to unique integer value.
- C. Set the `EmitDefaultValue` property of each data member to `false`.
- D. Set the `IsRequired` property of each data member to `true`.

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

`DataMemberAttribute.IsRequired` Property

Gets or sets a value that instructs the serialization engine whether the member must be present in the serialized data when it is being deserializing.

`DataMemberAttribute.IsRequired` Property

(<http://msdn.microsoft.com/en-us/library/system.runtime.serialization.datamemberattribute.isrequired.aspx>)

Data Contract Versioning

(<http://msdn.microsoft.com/en-us/library/ms731138%28v=VS.100%29.aspx>)

QUESTION 5

The following is an example of a SOAP envelope.

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope">
  <s:Header>
    <h:StoreId xmlns:h="http://www.contoso.com">6495</h:StoreId>
  </s:Header>
  <s:Body>
    <CheckStockRequest xmlns="http://www.contoso.com">
      <ItemId>2469</ItemId>
    </CheckStockRequest>
  </s:Body>
</s:Envelope>
```

You need to create a message contract that generates the SOAP envelope.

Which code segment should you use?

- A.

```
[MessageContract(WrapperName="http://www.contoso.com")]
public class CheckStockRequest
{
    [MessageHeader(Namespace="http://www.contoso.com")]
    public int StoreId { get; set; }

    [MessageBodyMember(Namespace="http://www.contoso.com")]
    public int ItemId { get; set; }
}
```
- B.

```
[MessageContract(WrapperNamespace="http://www.contoso.com")]
public class CheckStockRequest
{
    [MessageHeader(Namespace="http://www.contoso.com")]
    public int StoreId { get; set; }
```

```
    [MessageBodyMember(Namespace="http://www.contoso.com")]
    public int ItemId { get; set; }
}
```

C. [MessageContract(WrapperNamespace="http://www.contoso.com")]
public class CheckStockRequest
{
 [MessageHeader(Namespace="http://www.contoso.com")]
 public int StoreId { get; set; }
 public int ItemId { get; set; }
}

D. [MessageContract(WrapperNamespace="http://www.contoso.com")]
public class CheckStockRequest
{
 [MessageHeader(Namespace="http://www.contoso.com")]
 public int StoreId { get; set; }

 [MessageBodyMember]
 public int ItemId { get; set; }
}

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

MessageContractAttribute.WrapperNamespace Property

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.messagecontractattribute.wrappernamespace.aspx>)

I saw the answer D!!

QUESTION 6

You are developing a client that sends several types of SOAP messages to a Windows Communication Foundation (WCF) service method named PostData. PostData is currently defined as follows:

```
[OperationContract]
void PostData(Order data);
```

You need to modify PostData so that it can receive any SOAP message. Which code segment should you use?

- A. [OperationContract(IsOneWay=true, Action="*", ReplyAction="*")]
void PostData(Order data);
- B. [OperationContract(IsOneWay=true, Action="*", ReplyAction = "*")]
void PostData(BodyWriter data);
- C. [OperationContract]
void PostData(BodyWriter data);
- D. [OperationContract]
void PostData(Message data);

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

The Message class is fundamental to Windows Communication Foundation (WCF).

All communication between clients and services ultimately results in Message instances being sent and received.

You would not usually interact with the Message class directly. Instead, WCF service model constructs, such as data contracts, message contracts, and operation contracts, are used to describe incoming and outgoing messages.

However, in some advanced scenarios you can program using the Message class directly.

For example, you might want to use the Message class:

- When you need an alternative way of creating outgoing message contents (for example, creating a message directly from a file on disk) instead of serializing .NET Framework objects.
- When you need an alternative way of using incoming message contents (for example, when you want to apply an XSLT transformation to the raw XML contents) instead of deserializing into .NET Framework objects.
- When you need to deal with messages in a general way regardless of message contents (for example, when routing or forwarding messages when building a router, load-balancer, or a publish-subscribe system).

How to: Send and Receive a SOAP Message By Using the SoapClient and SoapService Classes

(<http://msdn.microsoft.com/en-us/library/aa529276.aspx>)

How to: Send and Receive a SOAP Message By Using the SoapSender and SoapReceiver Classes

(<http://msdn.microsoft.com/en-us/library/aa528828.aspx>)

QUESTION 7

A class named TestService implements the following interface:

```
[ServiceContract]
public interface ITestService
{
    [OperationContract]
    DateTime GetServiceTime();
}
```



<http://www.gratisexam.com/>

TestService is hosted in an ASP.NET application.

You need to modify the application to allow the GetServiceTime method to return the data formatted as JSON.

It must do this only when the request URL ends in /ServiceTime. What should you do?

- A. Add this attribute to the GetServiceTime method.

```
[WebInvoke(Method="POST")]
```

In the web.config file, add this element to system.serviceModel/behaviors/endpointBehaviors.

```
<behavior name="Json">
  <enableWebScript />
</behavior>
```

In the web.config file, configure TestService in the system.serviceModel/services collection as follows:

```
<service name="TestService">
  <endpoint address="/ServiceTime"
    contract="TestService"
    behaviorConfiguration="Json"
    binding="webHttpBinding" />
</service>
```

- B. Add this attribute to the GetServiceTime method.

```
[WebInvoke(Method="GET", UriTemplate="/ServiceTime",
ResponseFormat=WebMessageFormat.Json)]
```

In the web.config file, configure TestService in the system.serviceModel/services collection as follows:

```
<service name="TestService">
  <endpoint address="/ServiceTime"
    contract="TestService"
    binding="webHttpBinding" />
</service>
```

C. Add this attribute to the GetServiceTime method

```
[WebGet(ResponseFormat=WebMessageFormat.Json, UriTemplate="/ServiceTime")]
```

Create a new svc file named Jsonversion.svc with the following content.

```
<% @ServiceHost Service="TestService"
Factory="System.ServiceModel.ActivationWebServiceHostFactory" %>
```

D. Add this attribute to the GetServiceTime method.

```
[WebGet(UriTemplate="Json)/ServiceTime")]
```

Create a new .svc file named Jsonversion.svc with the following content

```
<% @ServiceHost Service="TestService"
Factory="System.ServiceModel.ActivationWebServiceHostFactory" %>
```

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

WebGetAttribute.ResponseFormat

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.web.webgetattribute.responseformat.aspx>)

The following example shows how to set the ResponseFormat property.

```
[OperationContract]
[WebGet(ResponseFormat= WebMessageFormat.Json)]
long Mod(long x, long y);
```

QUESTION 8

You are creating a Windows Communication Foundation (WCF) service that implements operations in a RESTful manner.

You need to add a delete operation. You implement the delete method as follows:

```
void DeleteItems(string id);
```

You need to configure WCF to call this method when the client calls the service with the HTTP DELETE operation. What should you do?

- A. Add the WebInvoke(UriTemplate="/Items/{id}", Method="DELETE") attribute to the operation
- B. Add the HttpDelete attribute to the operation
- C. Replace the string parameter with a RemovedActivityAction parameter
- D. Replace the return type with RemovedActivityAction.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

Advanced Web Programming

([http://msdn.microsoft.com/en-us/library/bb472541\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/bb472541(v=vs.90).aspx))

Example:


```

[ServiceContract]
public interface ICustomerCollection
{
    [OperationContract]
    [WebInvoke(Method = "POST", UriTemplate = "")]
    Customer AddCustomer(Customer customer);

    [OperationContract]
    [WebInvoke(Method = "DELETE", UriTemplate = "{id}")]
    void DeleteCustomer(string id);

    [OperationContract]
    [WebGet(UriTemplate = "{id}")]
    Customer GetCustomer(string id);

    [OperationContract]
    [WebGet(UriTemplate = "")]
    List<Customer> GetCustomers();

    [OperationContract]
    [WebInvoke(Method = "PUT", UriTemplate = "{id}")]
    Customer UpdateCustomer(string id, Customer newCustomer);
}

```

QUESTION 9

A Windows Communication Foundation (WCF) service implements a contract with one-way and request-reply operations.

The service is exposed over a TCP transport. Clients use a router to communicate with the service. The router is implemented as follows. (Line numbers are included for reference only.)

```

01 ServiceHost host = new ServiceHost(typeof(RoutingService));
02 host.AddServiceEndpoint(
03     typeof(ISimplexDatagramRouter),
04     new NetTcpBinding(), "net.tcp://localhost/Router"
05 );
06 List<ServiceEndpoint> lep = new List<ServiceEndpoint>();
07 lep.Add(
08     new ServiceEndpoint(
09         ContractDescription.GetContract(
10             typeof(ISimplexDatagramRouter)
11         ),
12     new NetTcpBinding(),
13     new EndpointAddress("net.tcp://localhost:8080/Logger")
14 );
15 );
16 RoutingConfiguration rc = new RoutingConfiguration();
17 rc.FilterTable.Add(new MatchAllMessageFilter(), lep);
18 host.Description.Behaviors.Add(new RoutingBehavior(rc));

```

Request-reply operations are failing. You need to ensure that the router can handle one-way and request-reply operations.

What should you do?

- A. Change line 03 as follows:

```
typeof(IRequestReplyRouter),
```
- B. Change line 03 as follows:

```
typeof(IDuplexSessionRouter),
```
- C. Change line 10 as follows:

```
typeof(IRequestReplyRouter)
```
- D. Change line 10 as follows:

```
typeof(IDuplexSessionRouter)
```

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

IDuplexSessionRouter Interface Defines the interface required to process messages from duplex session channels.

IDuplexSessionRouter Interface

([http://msdn.microsoft.com/en-us/library/system.servicemodel.routing.iduplexsessionrouter\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/system.servicemodel.routing.iduplexsessionrouter(v=vs.110).aspx))

QUESTION 10

You are modifying an existing Windows Communication Foundation (WCF) service that is defined as follows:

```
[ServiceContract]
public interface IMessageProcessor
{
    [OperationContract]
    void ProcessMessages();
}
public class MessageProcessor: IMessageProcessor
{
    public void ProcessMessage();
    SubmitOrder();
}
```

SubmitOrder makes a call to another service. The ProcessMessage method does not perform as expected under a heavy load.

You need to enable processing of multiple messages. New messages must only be processed when the ProcessMessage method is not processing requests, or when it is waiting for calls to SubmitOrder to return.

Which attribute should you apply to the MessageProcessor class?

- A. `CallbackBehavior(ConcurrencyMode=ConcurrencyMode.Reentrant)`
- B. `CallbackBehavior(ConcurrencyMode=ConcurrencyMode.Multiple)`
- C. `ServiceBehavior(ConcurrencyMode=ConcurrencyMode.Reentrant)`
- D. `ServiceBehavior(ConcurrencyMode=ConcurrencyMode.Multiple)`

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

ConcurrencyMode.Reentrant implies that the service (or callback) processes only one message at a given time (analogous to `ConcurrencyMode.Single`).

To ensure thread safety, Windows Communication Foundation (WCF) locks the `InstanceContext` processing a message so that no other messages can be processed.

In case of `Reentrant` mode, the `InstanceContext` is unlocked just before the service makes an outgoing call thereby allowing the subsequent call, (which can be reentrant as demonstrated in the sample) to get the lock next time it comes in to the service.

ConcurrencyMode Enumeration

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.concurrencyMode.aspx>)

QUESTION 11

A Windows Communication Foundation (WCF) service listens for messages at `net.tcp://www.contoso.com/MyService`.

It has a logical address at `http://www.contoso.com/MyService`. The configuration for the WCF client is as follows:

```
<endpoint address="http://www.contoso.com/MyService"
    binding="netTcpBinding"
```

```
bindingConfiguraton="NetTcpBinding_IMyService"  
contract="ServiceReference1.IMyService"  
name="NetTcpBinding_IMyService"/>
```

The generated configuration does not provide enough information for the client to communicate with the server.

You need to update the client so that it can communicate with the server. What should you do?

- A. In the client configuration, change the value of the address attribute to `net.tcp://www.contoso.com/MyService`
- B. In the client configuration, change the value of the address attribute to `net.tcp://www.contoso.com/MyService listen=http://www.contoso.com/MyService`.
- C. After instantiating the client and before invoking any service operation, add this line of code.
`EndpointBehaviors.Add(new EndpointDiscoveryBehavior(){ Enabled = true });`
- D. After instantiating the client and before invoking any service operation, add this line of code.
`client.Endpoint.Behaviors.Add(new ClientViaBehavior(new Uri("net.tcp://www.contoso.com/IMyService")));`

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

The **ClientViaBehavior** is used by clients to specify the URI for which the transport channel should be created.

If such a behavior exists in the behavior collection on a client endpoint, the transport uses the URI that it provides,

while all the other channel layers in the stack use the EndpointAddress provided at ChannelFactory construction time.

This EndpointAddress also becomes the WS-Addressing To header.

Use this behavior when the immediate network destination is not the intended processor of the message to enable multiple-hop

conversations when the calling application does not necessarily know the ultimate destination or when the destination Via header is not an address.

You can also set this value using the <clientVia> element in a client application configuration file.

ClientViaBehavior Class

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.description.clientviabehavior.aspx>)

QUESTION 12

A Windows Communication Foundation (WCF) service implements the following contract.

```
[ServiceContract]  
public interface IHelloService  
{  
    [OperationContract(WebGet(UriTemplate="hello?name={name}"))]  
    string SayHello(string name);  
}
```

The implementation is as follows:

```
public class HelloService: IHelloService  
{  
    public string SayHello(string name)  
    {  
        return "Hello " + name;  
    }  
}
```

```
}
```

The service is self-hosted, and the hosting code is as follows:

```
WebServiceHost svcHost = CreateHost();  
svcHost.Open();  
Console.ReadLine();  
svcHost.Close();
```

You need to implement CreateHost so that the service has a single endpoint hosted at `http://localhost:8000/HelloService`.

Which code segment should you use?

- A.

```
WebServiceHost svcHost = new WebServiceHost(typeof(HelloService));  
svcHost.AddServiceEndpoint(typeof(IHelloService),  
    new WebHttpBinding(WebHttpSecurityMode.None),  
    "http://localhost:8000/HelloService");  
return svcHost;
```
- B.

```
Uri baseAddress = new Uri("http://localhost:8000");  
WebServiceHost svcHost = new WebServiceHost(typeof(HelloService),  
    baseAddress);  
svcHost.AddServiceEndpoint(typeof(IHelloService),  
    new WebHttpBinding(WebHttpSecurityMode.None),  
    "HelloService");  
return svcHost;
```
- C.

```
WebServiceHost svcHost = new WebServiceHost(new HelloService());  
svcHost.AddServiceEndpoint(typeof(IHelloService),  
    new WebHttpBinding(WebHttpSecurityMode.None),  
    "http://localhost:8000/HelloService");  
returnn svcHost
```
- D.

```
Uri baseAddress = new Uri("http://localhost:8000/");  
WebServiceHost svcHost = new WebServiceHost(new HelloService(),  
    baseAddress);  
svcHost.AddServiceEndpoint(typeof(IHelloService),  
    new WebHttpBinding(WebHttpSecurityMode.None),  
    "HelloService");  
returnn svcHost;
```

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

WebServiceHost Class

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.web.webservicehost.aspx>)

Name	Description
WebServiceHost()	Initializes a new instance of the WebServiceHost class.
WebServiceHost(Object, Uri[])	Initializes a new instance of the WebServiceHost class with the specified singleton server instance and base address.
WebServiceHost(Type, Uri[])	Initializes a new instance of the WebServiceHost class with the specified service type and base address.

WebServiceHost.AddServiceEndpoint() Method :

Name	Description
AddServiceEndpoint	Adds the specified service endpoint to the hosted service. (Inherited from

(ServiceEndpoint)	ServiceHostBase.)
AddServiceEndpoint (String, Binding, String)	Adds a service endpoint to the hosted service with a specified contract, binding, and endpoint address. (Inherited from ServiceHostBase.)
AddServiceEndpoint (String, Binding, Uri)	Adds a service endpoint to the hosted service with a specified contract, binding, and a URI that contains the endpoint address. (Inherited from ServiceHostBase.)
AddServiceEndpoint (Type, Binding, String)	Adds a service endpoint to the hosted service with a specified contract, binding, and endpoint address. (Inherited from ServiceHost.)
AddServiceEndpoint (Type, Binding, Uri)	Adds a service endpoint to the hosted service with a specified contract, binding, and URI that contains the endpoint address. (Inherited from ServiceHost.)
AddServiceEndpoint (String, Binding, String, Uri)	Adds a service endpoint to the hosted service with a specified contract, binding, endpoint address and URI that contains the address at which it listens. (Inherited from ServiceHostBase.)
AddServiceEndpoint (String, Binding, Uri, Uri)	Adds a service endpoint to the hosted service with the specified contract, binding, and URIs that contain the endpoint and listening addresses. (Inherited from ServiceHostBase.)
AddServiceEndpoint (Type, Binding, String, Uri)	Adds a service endpoint to the hosted service with a specified contract, binding, an endpoint address, and a URI on which the service listens. (Inherited from ServiceHost.)
AddServiceEndpoint (Type, Binding, Uri, Uri)	Adds a service endpoint to the hosted service with a specified contract, binding, a URI that contains the endpoint address, and a URI on which the service listens. (Inherited from ServiceHost.)

WebServiceHost doesn't have a single param constructor.

```
public ServiceEndpoint AddServiceEndpoint(
    Type implementedContract,
    Binding binding,
    string address
)
```

Example:

```
static void Main(string[] args)
{
    Uri baseAddress = new Uri("http://localhost:8000/");

    WebServiceHost svcHost = new WebServiceHost(typeof(CalcService), baseAddress);

    try
    {
        svcHost.Open();

        Console.WriteLine("Service is running");
        Console.WriteLine("Press enter to quit...");
        Console.ReadLine();

        svcHost.Close();
    }
    catch (CommunicationException cex)
    {
        Console.WriteLine("An exception occurred: {0}", cex.Message);
        svcHost.Abort();
    }
}
```

QUESTION 13

You are building a client for a Windows Communication Foundation (WCF) service. You need to create a proxy to consume this service. Which class should you use?

- A. ChannelFactory<TChannel>
- B. ServiceHost
- C. ClientRuntime
- D. CommunicationObject

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

A factory that creates channels of different types that are used by clients to send messages to variously configured service endpoints.

ChannelFactory<TChannel> Class

(<http://msdn.microsoft.com/en-us/library/ms576132.aspx>)

QUESTION 14

A Windows Communication Foundation (WCF) service has a callback contract. You are developing a client application that will call this service.

You must ensure that the client application can interact with the WCF service. What should you do?

- A. On the OperationContractAttribute, set the AsyncPattern property value to true.
- B. On the OperationContractAttribute, set the ReplyAction property value to the endpoint address of the client.
- C. On the client, create a proxy derived from DuplexClientBase<TChannel>.
- D. On the client, use GetCallbackChannel<T>.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

DuplexClientBase<TChannel> Class Used to create a channel to a duplex service and associate that channel with a callback object.

DuplexClientBase<TChannel> Class

(<http://msdn.microsoft.com/en-us/library/ms576169.aspx>)

QUESTION 15

You are creating a Windows Communication Foundation (WCF) service. You have the following requirements:

- Messages must be sent over TCP
- The service must support transactions.
- Messages must be encoded using a binary encoding
- Messages must be secured using Windows stream-based security.

You need to implement a custom binding for the service. In which order should the binding stack be configured?

- A. tcpTransport, windowsStreamSecurity, transactionFlow, binaryMessageEncoding
- B. transactionFlow, binaryMessageEncoding, windowsStreamSecurity, tcpTransport
- C. windowsStreamSecurity, tcpTransport, binaryMessageEncoding, transactionFlow
- D. binaryMessageEncoding, transactionFlow, tcpTransport, windowsStreamSecurity

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

From higher to lower stack level

You can add binding elements by adding the desired BindingElement objects to its Elements collection. The order in which the binding element is added is very important. The order of adding the binding elements is as follows:

1. Transaction Flow (not required)
2. Reliable Messaging (not required)
3. Message Security (not required)
4. Composite Duplex (not required)
5. Message Encoding (required)
6. Transport Security (not required)
7. Transport (required)

Bindings

(<http://msdn.microsoft.com/en-us/library/ff650874.aspx>)

WCF Bindings in Depth

(<http://msdn.microsoft.com/en-us/magazine/cc163394.aspx>)

QUESTION 16

A Windows Communication Foundation (WCF) solution uses the following contracts. (Line numbers are included for reference only.)

```
01 [ServiceContract(CallbackContract=typeof(INameService))]
02 public interface IGreetingService
03 {
04     [OperationContract]
05     string GetMessage();
06 }
07
08 [ServiceContract]
09 public interface INameService
10 {
11     [OperationContract]
12     string GetName();
13 }
```

When the client calls GetMessage on the service interface, the service calls GetName on the client callback.

In the client, the class NameService implements the callback contract. The client channel is created as follows:

```
22 InstanceContext callbackContext = new InstanceContext(new NameService
23 ("client"));
24 ...
25 DuplexChannelFactory<IGreetingService> factory = new
DuplexChannelFactory<IGreetingService>(typeof(NameService), binding, address);
26 IGreetingService greetingService = factory.CreateChannel();
```

You need to ensure that the service callback is processed by the instance of NameService. What are two possible ways to achieve this goal? (Each correct answer presents a complete solution. Choose two.)

- A. Change line 25 to the following code segment:
`DuplexChannelFactory<IGreetingService> factory = new
DuplexChannelFactory<IGreetingService>(callbackContext, binding, address);`
- B. Change line 26 to the following code segment:
`IGreetingService greetingServicefactory = CreateChannel(callbackContext);`
- C. Add the following code segment after line 26:
`callbackContext.IncomingChannels.Add((IDuplexChannel)greetingService);`
- D. Add the following code segment after line 26:

```
callbackContext.OutgoingChannels.Add( ( IDuplexChannel )greetingService );
```

Correct Answer: AB

Section: (none)

Explanation

Explanation/Reference:

Example 1:

```
// Construct InstanceContext to handle messages on callback interface
InstanceContext instanceContext = new InstanceContext(new CallbackHandler());

// Create a client
CalculatorDuplexClient client = new CalculatorDuplexClient(instanceContext);
```

QUESTION 17

A Windows Communication Foundation (WCF) service implements the following contract. (Line numbers are included for reference only.)

```
01 [ServiceContract]
02 public interface IDataAccessService
03 {
04     [OperationContract]
05     void PutMessage(string message);
06     ...
07     [OperationContract]
08     [FaultContract(typeof(TimeoutFaultException))]
09     [FaultContract(typeof(FaultException))]
10     string SearchMessages(string search);
11 }
```

The implementation of the SearchMessages method throws TimeoutFaultException exceptions for database timeouts.

The implementation of the SearchMessages method also throws an Exception for any other issue it encounters while processing the request.

These exceptions are received on the client side as generic FaultException exceptions.

You need to implement the error handling code for SearchMessages and create a new channel on the client only if the channel faults.

What should you do?

- A. Catch and handle both TimeoutFaultException and FaultException.
- B. Catch both TimeoutFaultException and FaultException. Create a new channel in both cases.
- C. Catch and handle TimeoutFaultException. Catch FaultException and create a new channel.
- D. Catch and handle FaultException. Catch TimeoutFaultException and create a new channel.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 18

You are consuming a Windows Communication Foundation (WCF) service in an ASP. NET Web application.

The service interface is defined as follows:

```
[ServiceContract]
public interface ICatalog
{
    [OperationContract]
    [WebGet(UriTemplate="/Catalog/Items/{id}"),
```



```

ResponseFormat=WebMessageFormat.Json)]
    string RetrieveItemDescription(int id);
}

```

The service is hosted at Catalogsvc.

You need to call the service using jQuery to retrieve the description of an item as indicated by a variable named itemId.

Which code segment should you use?

- A.

```
$get(String.format("/Catalogsvc/Catalog/Items/id{0}", itemId) null, function (data) {
    ...
}, "javascript");
```
- B.

```
$get(String.format("/Catalogsvc/Catalog/Items/{0}", itemId), null, function (data) {
    ...
}, "json");
```
- C.

```
$get(String.format("/Catalogsvc/Catalog/Items/{0}", itemId), null, function (data) {
    ...
}, "xml");
```
- D.

```
$get(String.format("/Catalogsvc/Catalog/Items/id{0}", itemId), null, function (data) {
    ...
}, "json");
```

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 19

You are consuming a Windows Communication Foundation (WCF) service.

The service interface is defined as follows:

```

[DataContract(Namespace = '')]
public class Item
{
}

[ServiceContract(Namespace = '')]
public interface Catalog
{
    [OperationContract]
    [WebInvoke(Method="POST", UriTemplate="{Item}")]
    Item UpdateItem(Item item);
}

```

The client application receives a WebResponse named response with the response from the service. You need to deserialize this response into a strongly typed object representing the return value of the method.

Which code segment should you use?

- A.

```
DataContractSerializer s = new DataContractSerializer(typeof(Item));
Item item = s.ReadObject(response.GetResponseStream()) as Item;
```
- B.

```
BinaryFormatter f = new BinaryFormatter();
```

```
Item item = f.Deserialize(response.GetResponseStream() as Item;
```

- C. `XmlDictionaryReader r = JsonSerializerFactory.CreateJsonReader(response.GetResponseStream(), XmlDictionaryReaderQuotas.Max);
DataContractSerializer s = new DataContractSerializer(typeof(Item));
Item item = s.ReadObject(r) as Item;`
- D. `DataContractJsonSerializer s = new DataContractJsonSerializer(typeof(Item));
Item item = s.ReadObject(response.GetResponseStream()) as Item;`

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

Use the `DataContractSerializer` class to serialize and deserialize instances of a type into an XML stream or document.

For example, you can create a type named `Person` with properties that contain essential data, such as a name and address.

You can then create and manipulate an instance of the `Person` class and write all of its property values in an XML document for later retrieval,

or in an XML stream for immediate transport. Most important, the `DataContractSerializer` is used to serialize and deserialize data sent in messages.

Apply the `DataContractAttribute` attribute to classes, and the `DataMemberAttribute` attribute to class members to specify properties and fields that are serialized.

Serialization and Deserialization

(<http://msdn.microsoft.com/en-us/library/ms731073%28v=VS.100%29.aspx>)

QUESTION 20

A Windows Communication Foundation (WCF) application exposes a service as a SOAP endpoint for consumption by cross-platform clients.

During integration testing, you find that one of the clients is not generating the correct messages to the WCF application.

In order to debug the issue and fix the communication, you need to configure the service to log messages received from the client.

What should you do?

- A. Set an `etwTracking` behavior on the service and configure a listener for the `System.ServiceModel` trace source.
- B. Set an `etwTracking` behavior on the service and configure a listener for the `System.ServiceModel.MessageLogging` trace source.
- C. Enable `messageLogging` in the `System.ServiceModel` diagnostics element configuration and configure a listener for the `System.ServiceModel.MessageLogging` trace source.
- D. Enable `messageLogging` in the `System.ServiceModel` diagnostics element configuration and configure a listener for the `System.ServiceModel` trace source.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

Configuring Message Logging

(<http://msdn.microsoft.com/en-us/library/ms730064.aspx>)

Example:

```
<system.diagnostics>  
  <sources>  
    <source name="System.ServiceModel.MessageLogging">  
      <listeners>  
        <add name="messages"</add name="messages">  
      </listeners>  
    </source>  
  </sources>  
</system.diagnostics>
```

```

        type="System.Diagnostics.XmlWriterTraceListener"
        initializeData="c:\logs\messages.svclog" />
    </listeners>
</source>
</sources>
</system.diagnostics>

<system.serviceModel>
  <diagnostics>
    <messageLogging
      logEntireMessage="true"
      logMalformedMessages="false"
      logMessagesAtServiceLevel="true"
      logMessagesAtTransportLevel="false"
      maxMessagesToLog="3000"
      maxSizeOfMessageToLog="2000" />
    </diagnostics>
  </system.serviceModel>

```

QUESTION 21

A Windows Communication Foundation (WCF) client application is consuming an RSS syndication feed from a blog.

You have a SyndicationFeed variable named feed. The application iterates through the items as follows. (Line numbers are included for reference only.)

```

01 foreach (SyndicationItem item in feed.Items)
02 {
03 }

```

You need to display the content type and body of every syndication item to the console. Which two lines of code should you insert between lines 02 and 03?

- A. `Console.WriteLine(item.Content.Type);`
`Console.WriteLine(((TextSyndicationContent)item.Content).Text);`
- B. `Console.WriteLine(item.Content.GetType());`
`Console.WriteLine(((TextSyndicationContent)item.Content).Text);`
- C. `Console.WriteLine(item.Content.Type);`
`Console.WriteLine(item.Content.ToString());`
- D. `Console.WriteLine(item.Content.GetType());`
`Console.WriteLine(item.Content.ToString());`

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

SyndicationItem Class Represents a feed item, for example an RSS <item> or an Atom <entry>.

TextSyndicationContent Represents any SyndicationItem content intended to be displayed to an end user. Properties:

AttributeExtensions Gets the attribute extensions for this content. (Inherited from SyndicationContent.)

Text Gets the text of the TextSyndicationContent.

Type Gets the content type of the TextSyndicationContent. (Overrides SyndicationContent.Type.)

SyndicationItem Class

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.syndication.syndicationitem.aspx>)

4.1 ATOM 1.0 Formatted Feed

([http://msdn.microsoft.com/en-us/library/ee682805\(v=office.12\).aspx](http://msdn.microsoft.com/en-us/library/ee682805(v=office.12).aspx))

4.2 RSS 2.0 Formatted Feed

([http://msdn.microsoft.com/en-us/library/ee682799\(v=office.12\).aspx](http://msdn.microsoft.com/en-us/library/ee682799(v=office.12).aspx))

QUESTION 22

You are creating a Windows Communication Foundation (WCF) service to process orders. The data contract for the order is defined as follows:

```
[DataContract]
public class Order
{
    [DataMember]
    public string CardHolderName { get; set; }
    [DataMember]
    public string CreditCardNumber { get; set; }
}
```

You have the following requirements:

- Enable the transmission of the contents of Order from the clients to the service.
- Ensure that the contents of CreditCardNumber are not sent across the network in clear text.
- Ensure that the contents of CreditCardNumber are accessible by the service to process the order.

You need to implement the service to meet these requirements. What should you do?

- A. Add a DataProtectionPermission attribute to the CreditCardNumber property and set the ProtectData property to true.
- B. Convert the DataContract to a MessageContract and set the ProtectionLevel property to SignAndEncrypt.
- C. Change the data type of CreditCardNumber from string to SecureString.
- D. Implement the CreditCardNumber property getter and setter. In the setter, run the value of the CreditCardNumber through the MD5CryptoServiceProvider class TransformBlock method.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

A message contract can indicate whether the headers and/or body of the message should be digitally signed and encrypted.

This is done by setting the System.ServiceModel.MessageContractMemberAttribute.ProtectionLevel property on the MessageHeaderAttribute and MessageBodyMemberAttribute attributes.

The property is an enumeration of the System.Net.Security.ProtectionLevel type and can be set to None (no encryption or signature), Sign (digital signature only), or EncryptAndSign (both encryption and a digital signature). The default is EncryptAndSign.

Using Message Contracts

(<http://msdn.microsoft.com/en-us/library/ms730255.aspx>)

QUESTION 23

A Windows Communication Foundation (WCF) client uses the following service contract. (Line numbers are included for reference only.)

```
01 [ServiceContract]
02 public interface IService
03 {
04     [OperationContract]
05     string Operation1();
06     [OperationContract]
07     string Operation2();
08 }
```

You need to ensure that all calls to Operation1 and Operation2 from the client are encrypted and signed. What should you do?

- A. Set the ProtectionLevel property in line 01 to EncryptAndSign.
- B. Set the ProtectionLevel property in line 04 and line 06 to Sign.
- C. Add a SecurityCriticalAttribute for each operation.
- D. Add a SecuritySafeCriticalAttribute for each operation.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

How to set ProtectionLevel Property

(<http://msdn.microsoft.com/en-us/library/aa347791.aspx>)

The ProtectionLevel property is found on many different classes, such as the ServiceContractAttribute and the OperationContractAttribute classes.

The property controls how a part (or whole) of a message is protected. To understand the protection level feature, the following basic statements apply:

Three basic levels of protection exist for any part of a message. The property (wherever it occurs) is set to one of the ProtectionLevel enumeration values.

In ascending order of protection, they include:

None

Sign The protected part is digitally signed. This ensures detection of any tampering with the protected message part.

EncryptAndSign The message part is encrypted to ensure confidentiality before it is signed.

You can set protection requirements only for application data with this feature.

For example, WS-Addressing headers are infrastructure data and, therefore, are not affected by the ProtectionLevel.

When the security mode is set to Transport, the entire message is protected by the transport mechanism. Therefore, setting a separate protection level for different parts of a message has no effect.

The ProtectionLevel is a way for the developer to set the minimum level that a binding must comply with. When a service is deployed, the actual binding specified in configuration may or may not support the minimum level.

For example, by default, the BasicHttpBinding class does not supply security (although it can be enabled). Therefore, using it with a contract that has any setting other than None will cause an exception to be thrown.

If the service requires that the minimum ProtectionLevel for all messages is Sign, a client (perhaps created by a non-WCF technology)

can encrypt and sign all messages (which is more than the minimum required). In this case, WCF will not throw an exception

because the client has done more than the minimum. Note, however, that WCF applications (services or clients) will not over-secure a message part

if possible but will comply with the minimum level. Also note that when using Transport as the security mode, the transport may over-secure

the message stream because it is inherently unable to secure at a more granular level.

If you set the ProtectionLevel explicitly to either Sign or EncryptAndSign, then you must use a binding with security enabled or an exception will be thrown.

If you select a binding that enables security and you do not set the ProtectionLevel property anywhere on the contract, all application data will be encrypted and signed.

If you select a binding that does not have security enabled (for example, the BasicHttpBinding class has security disabled by default),

and the ProtectionLevel is not explicitly set, then none of the application data will be protected.

If you are using a binding that applies security at the transport level, all application data will be secured according to the capabilities of the transport.

If you use a binding that applies security at the message level, then application data will be secured according to the protection levels set on the contract.
If you do not specify a protection level, then all application data in the messages will be encrypted and signed.

The ProtectionLevel can be set at different scoping levels.

QUESTION 24

You are developing a Windows Communication Foundation (WCF) service. The service needs to access out-of-process resources.

You need to ensure that the service accesses these resources on behalf of the originating caller. What should you do?

- A. Set the value of ServiceSecurityContext.Current.WindowsIdentity.ImpersonationLevel to TokenImpersonationLevel.Impersonation
- B. Set the value of ServiceSecurityContext.Current.WindowsIdentity.ImpersonationLevel to TokenImpersonationLevel.Delegation
- C. Set the PrincipalPermissionAttribute on the service contract and update the binding attribute in the endpoint element of the configuration file to wsHttpBinding
- D. Set the PnnicipalPermissionAttribute on the service contract and update the bindingConfiguration attribute in the endpoint element of the configuration file to wsHttpBinding

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

Impersonation is a common technique that WCF services use to assume the original caller's identity in order to authorize access to service resources (such as files or database tables). Service resources can be resources that are either local to the service machine or remotely hosted. Impersonation is used to access resources on the same machine as the service, while delegation is used to access resources that are remotely hosted.

By default, impersonation is disabled and resources are accessed by using the WCF service's process identity.

Impersonation allows you to access local resources and perform other operations using the authenticated user's identity

or a specific Windows identity. You can enable impersonation either programmatically or by applying appropriate attributes at operation or service levels.

You can impersonate imperatively or declaratively. Imperative impersonation is performed programmatically at run time and can vary depending

on business logic or other conditions. Declarative impersonation is applied with a static attribute that can be associated with an operation or an entire interface.

In general, you should use imperative impersonation when you need the fine granularity made possible by writing the impersonation logic into your code.

If you do not need such fine granularity, you can use declarative impersonation.

Delegation allows you to use an impersonation token to access network resources. Your ability to use delegation depends

on the authentication mechanism in use and appropriate account configuration.

Delegation and Impersonation with WCF

(<http://msdn.microsoft.com/en-us/library/ms730088%28v=VS.90%29.aspx>)

QUESTION 25

A Windows Communication Foundation (WCF) service that handles corporate accounting must be changed

to comply with government regulations of auditing and accountability. You need to configure the WCF service to execute under the Windows logged-on identity of the calling application. What should you do?

- A. Within the service configuration, add a ServiceAuthorization behavior to the service, and set ImpersonateCallerForAllOperations to true.
- B. Within the service configuration, add a ServiceAuthenticationManager behavior to the service, and set ServiceAuthenticationManagerType to Impersonate.
- C. Within the service configuration, add a serviceSecurityAudit behavior to the service, and set serviceAuthorizationAuditLevel to SuccessOrFailure.
- D. Within the service configuration, add a ServiceCredentials behavior to the service, and set type to Impersonate.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

```
// Code to create a ServiceHost not shown.  
ServiceAuthorizationBehavior MyServiceAuthoriationBehavior =  
serviceHost.Description.Behaviors.Find<ServiceAuthorizationBehavior>();  
MyServiceAuthoriationBehavior.ImpersonateCallerForAllOperations = true;
```

Delegation and Impersonation with WCF

(<http://msdn.microsoft.com/en-us/library/ms730088%28v=VS.90%29.aspx>)

QUESTION 26

You have a secured Windows Communication Foundation (WCF) service. You need to track unsuccessful attempts to access the service. What should you do?

- A. Set the authorizationManagerType attribute of the serviceAuthorization behavior to Message.
- B. Set the includeExceptionDetailsInFaults attribute of the serviceDebug behavior to true.
- C. Set the Mode attribute of the security configuration element to Message.
- D. Set the messageAuthenticationAuditLevel attribute of the serviceSecurityAudit behavior to Failure.

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

<ServiceSecurityAudit>

Specifies settings that enable auditing of security events during service operations.

The AuditLevel is an enumeration with the following members:

Failure : Only failed security-related events are recorded.

None : No events are recorded.

Success : Only successful security-related events are recorded.

SuccessOrFailure : Both failed and successful security-related events are recorded.

<ServiceSecurityAudit>

(<http://msdn.microsoft.com/en-us/library/ms731694.aspx>)

QUESTION 27

A Windows Communication Foundation (WCF) solution uses the following contract to share a message across its clients.

(Line numbers are included for reference only.)

```

01 [ServiceContract]
02 public interface ITeamMessageService
03 {
04     [OperationContract]
05     string GetMessage();
06     [OperationContract]
07     void PutMessage(string message);
08 }
09 }

```

The code for the service class is as follows:

```

10 public class TeamMessageService: ITeamMessageService
11 {
12     Guid key = Guid.NewGuid();
13     string message = "Today's Message";
14     public string GetMessage()
15     {
16         return string.Format("Message:{0} Key:{1}",
17             message, key);
18     }
19     public void PutMessage(string message)
20     {
21         this.message = message;
22     }
23 }

```

The service is self-hosted. The hosting code is as follows:

```

24 ServiceHost host = new ServiceHost(typeof(TeamMessageService));
25 BasicHttpBinding binding = new BasicHttpBinding(BasicHttpSecurityMode.None);
26 host.AddServiceEndpoint(MyApplication.ITeamMessageService, binding, "http://
localhost:12345");
27 host.Open();

```

You need to ensure that all clients calling GetMessage will retrieve the same string, even if the message is updated by clients calling PutMessage.

What should you do

- A. Add the following attribute to the TeamMessageService class, before line 10.
[ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]
- B. Add the following attribute to the TeamMessageService class, before line 10.
[ServiceBehavior(InstanceContextMode = InstanceContextMode.PerSession)]

Then change the binding definition on the service at line 25, and on the client to the following

```

WSHttpBinding binding = new WSHttpBinding(SecurityMode.None);
binding.ReliableSession.Enabled = true;

```

- C. Pass a service instance to the instancing code in line 24, as follows.
ServiceHost host = new ServiceHost(new TeamMessageService());
- D. Redefine the message string in line 13, as follows
static string message = "Today's Message";

Then change the implementation of PutMessage in lines 19-22 to the following

```

public void PutMessage(string message)
{
    TeamMessageServiceMessage.PutMessage;
}

```

Correct Answer: A
Section: (none)

Explanation

Explanation/Reference:

InstanceContextMode Enumeration

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.instancecontextmode.aspx>)

PerSession A new InstanceContext object is created for each session.

PerCall A new InstanceContext object is created prior to and recycled subsequent to each call. If the channel does not create a session this value behaves as if it were PerCall.

Single Only one InstanceContext object is used for all incoming calls and is not recycled subsequent to the calls. If a service object does not exist, one is created.

QUESTION 28

A Windows Communication Foundation (WCF) solution exposes the following service over a TCP binding. (Line numbers are included for reference only.)

```
01 [ServiceContract]
02 [ServiceBehavior(ConcurrencyMode = ConcurrencyMode.Multiple)]
03 public class DataAccessService
04 {
05     [OperationContract]
06     public void PutMessage(string message)
07     {
08         MessageDatabase.PutMessage(message);
09     }
10     [OperationContract]
11     public string[] SearchMessages(string search)
12     {
13         return MessageDatabase.SearchMessages(search);
14     }
15 }
```

MessageDatabase supports a limited number of concurrent executions of its methods.

You need to change the service to allow up to the maximum number of executions of the methods of MessageDatabase.

This should be implemented without preventing customers from connecting to the service. What should you do?

- A. Change the service behavior as follows.
[ServiceBehavior(ConcurrencyMode = ConcurrencyMode.Multiple,
InstanceContextMode = InstanceContextMode.Single)]
- B. Change the service behavior as follows.
[ServiceBehavior(ConcurrencyMode = ConcurrencyMode.Single,
InstanceContextMode = InstanceContextMode.PerSession)]
- C. Add a throttling behavior to the service, and configure the maxConcurrentCalls.
- D. Add a throttling behavior to the service, and configure the maxConcurrentSessions.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

ServiceThrottlingBehavior Class

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.description.servicethrottlingbehavior.aspx>)

ServiceThrottlingBehavior Properties

MaxConcurrentCalls Gets or sets a value that specifies the maximum number of messages actively processing across a ServiceHost.

MaxConcurrentInstances Gets or sets a value that specifies the maximum number of InstanceContext objects in the service that can execute at one time.

MaxConcurrentSessions Gets or sets a value that specifies the maximum number of sessions a ServiceHost object can accept at one time.

The MaxConcurrentCalls property specifies the maximum number of messages actively processing across a ServiceHost object. Each channel can have one pending message that does not count against the value of MaxConcurrentCalls until begins to process it.

QUESTION 29

A Windows Communication Foundation (WCF) solution provides a session-based counter. The service is self-hosted. The hosting code is as follows:

```
ServiceHost host = new ServiceHost(typeof(CounterService));  
NetTcpBinding binding1 = new NetTcpBinding(SecurityMode.None);  
host.AddServiceEndpoint("MyApplication.ICounterService", binding1, "net.tcp://  
localhost:23456");  
host.Open();
```

This service is currently exposed over TCP, but needs to be exposed to external clients over HTTP. Therefore, a new service endpoint is created with the following code:

```
host.AddServiceEndpoint("MyApplication.ICounterService", binding2, "http://  
localhost:12345");
```

You need to complete the implementation and ensure that the session-based counter will perform over HTTP as it does over TCP. What should you do?

A. Define binding2 as follows:

```
WS2007HttpBinding binding2 = new WS2007HttpBinding(SecurityMode.None);
```

Configure binding2 as follows:

```
binding2.ReliableSession.Enabled = true;
```

B. Define binding2 as follows:

```
WSHttpBinding binding2 = new WSHttpBinding(SecurityMode.None);
```

Add the following behavior to the service implementation:

```
[ServiceBehavior(InstanceContextMode = InstanceContextMode.PerSession)]
```

C. Define binding2 as follows:

```
BasicHttpBinding binding2 = new BasicHttpBinding  
(BasicHttpSecurityMode.None);
```

Enable cookies for binding2.

```
binding2.AllowCookies = true;
```

D. Define binding2 as follows:

```
BasicHttpBinding binding2 = new BasicHttpBinding  
(BasicHttpSecurityMode.None);
```

Add the following behavior to the service implementation:

```
[ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]
```

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

WS2007HttpBinding Class

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.ws2007httpbinding.aspx>)

WS2007HttpBinding Class

Represents an interoperable binding that derives from `WSHttpBinding` and provides support for the updated versions of the `Security`, `ReliableSession`, and `TransactionFlow` binding elements.

The `WS2007HttpBinding` class adds a system-provided binding similar to `WSHttpBinding` but uses the Organization for the Advancement of Structured Information Standards (OASIS) standard versions of the `ReliableSession`, `Security`, and `TransactionFlow` protocols. No changes to the object model or default settings are required when using this binding.

QUESTION 30

A Windows Communication Foundation (WCF) solution uses the following contract:

```
[ServiceContract(SessionMode=SessionMode.Allowed)]
public interface IMyService
{
    [OperationContract(IsTerminating=false)]
    void Initialize();
    [OperationContract(IsInitiating=false)]
    void DoSomething();
    [OperationContract(IsTerminating=true)]
    void Terminate();
}
```

You need to change this interface so that:

- `Initialize` is allowed to be called at any time before `Terminate` is called.
- `DoSomething` is allowed to be called only after `Initialize` is called, and not allowed to be called after `Terminate` is called.
- `Terminate` will be allowed to be called only after `Initialize` is called.

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

- A. Change the `ServiceContract` attribute of the `IMyService` interface to the following.
`[ServiceContract(SessionMode=SessionMode.Required)]`
- B. Change the `ServiceContract` attribute of the `IMyService` interface to the following.
`[ServiceContract(SessionMode=SessionMode.Allowed)]`
- C. Change the `OperationContract` attribute of the `Initialize` operation to the following.
`[OperationContract(IsInitiating = true, IsTerminating = false)]`
- D. Change the `OperationContract` attribute of the `Terminate` operation to the following.
`[OperationContract(IsInitiating = false, IsTerminating = true)]`

Correct Answer: AD

Section: (none)

Explanation

Explanation/Reference:

OperationContractAttribute.IsInitiating

Gets or sets a value that indicates whether the method implements an operation that can initiate a session on the server (if such a session exists).

OperationContractAttribute.IsInitiating Property

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.operationcontractattribute.isinitiating.aspx>)

Example:

The following example is a service that implements a service contract that specifies three methods. The service requires a session. If a caller's first call is to any operation other than `MethodOne`, the channel is refused and an exception is thrown. When a caller initiates a session by calling `MethodOne`, that caller can terminate the communication session at any time by calling `MethodThree`. `MethodTwo` can be called any number of times during a session.

```

C#
[ServiceContract(SessionMode=SessionMode.Required)]
public class InitializeAndTerminateService
{
    [OperationContract(IsOneWay=true, IsInitiating=true, IsTerminating=false)]
    public void MethodOne()
    {
        return;
    }

    [OperationContract(IsInitiating=false, IsTerminating=false)]
    public int MethodTwo(int x, out int y)
    {
        y = 34;
        return 0;
    }

    [OperationContract(IsOneWay=true, IsInitiating=false, IsTerminating=true)]
    public void MethodThree()
    {
        return;
    }
}

```

QUESTION 31

A Windows Communication Foundation (WCF) client and service share the following service contract interface:

```

[ServiceContract]
public interface IContosoService
{
    [OperationContract]
    void SavePerson(Person person);
}

```

They also use the following binding:

```

NetTcpBinding binding = new NetTcpBinding() { TransactionFlow = true };

```

The client calls the service with the following code:

```

using (TransactionScope ts = new TransactionScope
(TransactionScopeOption.Required))
{
    IContosoService client = factory.CreateChannel();
    client.SavePerson(person);
    Console.WriteLine
(Transaction.Current.TransactionInformation.DistributedIdentifier);
    ts.Complete();
}

```

The service has the following implementation for SavePerson:

```

public void IContosoService.SavePerson(Person person)
{
    person.Save();
    Console.WriteLine
(Transaction.Current.TransactionInformation.DistributedIdentifier);
}

```

The distributed identifiers do not match on the client and the server.

You need to ensure that the client and server enlist in the same distributed transaction. What should you do?

A. Add the following attributes to the SavePerson operation on IContosoService.

```

[OperationBehavior(TransactionScope.Required = true)]
[TransactionFlow(TransactionFlowOption.Mandatory)]

```

B. Add the following attributes to the SavePerson operation on IContosoService
[TransactionFlow(TransactionFlowOption.Mandatory)]
[OperationBehavior(TransactionScope.Required = true)]

C. Add the following attribute to the SavePerson operation on IContosoService
[OperationBehavior(TransactionScope.Required = true)]

Add the following attribute to the implementation of SavePerson.
[TransactionFlow(TransactionFlowOption.Allowed)]

D. Add the following attribute to the SavePerson operation on IContosoService
[TransactionFlow(TransactionFlowOption.Allowed)]

Add the following attribute to the implementation of SavePerson.
[OperationBehavior(TransactionScope.Required = true)]

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

The TransactionFlowAttribute is an attribute used declaratively to associate a specific transaction flow policy with a service operation.

The TransactionFlowOption property of this attribute specifies whether the respective operation accepts a transaction flowed from the client,

or if the operation requires the client to always flow a transaction. The TransactionFlowAttribute can also be used as an operation behavior

to programmatically associate a transaction flow policy with a specific operation. In this case, it should be added to the Behaviors

collection on the operation's description.

Transaction Flow Settings

Transaction flow settings are generated for a service endpoint as a result of the intersection of the following three values:

The TransactionFlowAttribute attribute specified for each method in the service contract.

The TransactionFlow binding property in the specific binding.

The TransactionFlowProtocol binding property in the specific binding. The TransactionFlowProtocol binding property enables you to choose

among two different transaction protocols that you can use to flow a transaction. The following sections briefly describe each of them.

How to: Create a Transactional Service

(<http://msdn.microsoft.com/en-us/library/ms730232.aspx>)

QUESTION 32

A service implements the following contract. (Line numbers are included for reference only)

```
01 [ServiceContract(SessionMode = SessionMode.Required)]
02 public interface IContosoService
03 {
04     [OperationContract(IsOneWay = true, IsInitiating = true)]
05     void OperationOne(string value);
06
07     [OperationContract(IsOneWay = true, IsInitiating = false)]
08     void OperationTwo(string value);
09 }
```

The service is implemented as follows:

```
10 class ContosoService: IContosoService
11 {
12     public void OperationOne(string value) {...}
13     ...
14     public void OperationTwo(string value) {...}
```

15 }

ContosoService uses NetMsmqBinding to listen for messages. The queue was set up to use transactions for adding and removing messages.

You need to ensure that OperationOne and OperationTwo execute under the same transaction context when they are invoked in the same session.

What should you do?

- A. Insert the following attribute to OperationOne on IContosoService
[TransactionFlow(TransactionFlowOption.Mandatory)]

Insert the following attribute to OperationTwo on IContosoService
[TransactionFlow(TransactionFlowOption.Mandatory)]

- B. Insert the following attribute to OperationOne on ContosoService
[OperationBehavior(TransactionScopeRequired=true,
TransactionAutoComplete=false)]

Insert the following attribute to OperationTwo on ContosoService.
[OperationBehavior(TransactionScopeRequired=true,
TransactionAutoComplete=true)]

- C. Add the following XML segment to the application config file in the system serviceModel/bindings configuration section

```
<netMsmqBinding>  
  <binding name="contosoTx" durable="true" receiveContextEnabled="true" />  
</netMsmqBinding>
```

Then use the NetMsmqBinding named contosoTx to listen for messages from the clients.

- D. Add the following XML segment to the application config file in the systemserviceModel/bindings configuration section.

```
<customBinding>  
  <binding name="contosoTx">  
    <transactionFlow />  
    <binaryMessageEncoding />  
    <msmqTransport durable="true" />  
  </binding>  
</customBinding>
```

Then use the CustommiBinding named contosoTx to listen fcw messages from the clients.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

The OperationBehaviorAttribute attribute is a programming model feature that enables common features that developers otherwise must implement themselves.

The AutoDisposeParameters property controls whether parameter objects passed to an operation are disposed when the operation completes.

The TransactionAutoComplete property specifies whether the transaction in which the method executes is automatically committed if no unhandled exceptions occur.

The TransactionScopeRequired property specifies whether a method must execute within a transaction.

The Impersonation property specifies whether the service operation can, must, or cannot impersonate the caller's identity.

The ReleaseInstanceMode property specifies when service objects are recycled during the method invocation process.

TransactionFlowOption is a ServiceBehavior feature

ServiceModel Transaction Attributes

([http://msdn.microsoft.com/en-us/library/ms730250\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/ms730250(v=vs.90).aspx))

QUESTION 33

A WCF service code is implemented as follows. (Line numbers are included for reference only)

```
01 [ServiceContract]
02 [ServiceBehavior(InstanceContextMode =
03 InstanceContextMode.Single)]
04 public class CalculatorService
05 {
06     [OperationContract]
07     public double Calculate(double op1, string op, double op2)
08     {
09     }
10 }
```

You need to increase the rate by which clients get the required response from the service. What are two possible ways to achieve this goal? (Each correct answer presents a complete solution. Choose two.)

- A. Change the service behavior to the following:
[ServiceBehavior(InstanceContextMode=InstanceContextMode.Single, ConcurrencyMode=ConcurrencyMode.Multiple)]
- B. Change the service behavior to the following.
[ServiceBehavior(InstanceContextMode=InstanceContextMode.PerCall)]
- C. Require the clients use threads, the Parallel Task Library, or other mechanism to issue service calls in parallel.
- D. Require the clients to use async operations when calling the service.

Correct Answer: AB

Section: (none)

Explanation

Explanation/Reference:

The ConcurrencyMode property interacts with some other settings. For example, if the InstanceContextMode value is set to Single the result is that your service can only process one message at a time unless you also set the ConcurrencyMode value to Multiple.

InstanceContextMode Enumeration

PerSession A new InstanceContext object is created for each session.

PerCall A new InstanceContext object is created prior to and recycled subsequent to each call. If the channel does not create a session this value behaves as if it were PerCall.

Single Only one InstanceContext object is used for all incoming calls and is not recycled subsequent to the calls. If a service object does not exist, one is created.

Note For singleton lifetime behavior (for example, if the host application calls the ServiceHost constructor and passes an object to use as the service), the service class must set InstanceContextMode to InstanceContextMode.Single, or an exception is thrown when the service host is opened.

ConcurrencyMode Enumeration

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.concurrencymode.aspx>)

QUESTION 34

A Windows Communication Foundation (WCF) solution uses the following contracts. (Line numbers are included for reference only.)

```
01 [ServiceContract(CallbackContract=typeof(INameService))]
02 public interface IGreetingService
03 {
```

```

04     [OperationContract]
05     string GetMessage();
06 }
07 [ServiceContract]
08 public interface INameService
09 {
10     [OperationContract]
11     string GetName();
12 }

```

The code that implements the IGreetingService interface is as follows:

```

20 public class GreetingService : IGreetingService
21 {
22     public string GetMessage()
23     {
24         INameService clientChannel =
OperationContext.Current.GetCallbackChannel<INameService>();
25         string clientName = clientChannel.GetName();
26         return String.Format("Hello {0}", clientName);
27     }
28 }

```

The service is self-hosted. The hosting code is as follows:

```

30 ServiceHost host = new ServiceHost(typeof(GreetingService));
31 NetTcpBinding binding = new NetTcpBinding(SecurityMode.None);
32 host.AddServiceEndpoint("MyApplication.IGreetingService", binding,
"net.tcp://localhost:12345");
33 host.Open();

```

The code that implements the INameService interface is as follows:

```

40 class NameService : INameService
41 {
42     string name;
43     public NameService(string name)
44     {
45         this.name = name;
46     }
47     public string GetName()
48     {
49         return name;
50     }
51 }

```

Currently, this code fails at runtime, and an `InvalidOperationException` is thrown at line 25. You need to correct the code so that the call from the service back to the client completes successfully. What are two possible ways to achieve this goal? (Each correct answer presents a complete solution. Choose two.)

- A. Change the service contract definition in line 01 as follows.
`[ServiceContract(CallbackContract = typeof(INameService), SessionMode = SessionMode.Required)]`
- B. Add the following attribute to the NameService class, before line 40.
`[ServiceBehavior(ConcurrencyMode = ConcurrencyMode.Reentrant)]`
- C. Add the following attribute to the GreetingService class, before line 20.
`[ServiceBehavior(ConcurrencyMode = ConcurrencyMode.Reentrant)]`
- D. Add the following attribute to the GreetingService class, before line 20.
`[ServiceBehavior(ConcurrencyMode = ConcurrencyMode.Multiple)]`

Correct Answer: CD

Section: (none)

Explanation

Explanation/Reference:

ConcurrencyMode Enumeration

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.concurrency.mode.aspx>)

QUESTION 35

A Windows Communication Foundation (WCF) service has the following contract:

```
[ServiceContract]
public class ContosoService
{
    [OperationContract]
    [TransactionFlow(TransactionFlowOption.Mandatory)]
    [OperationBehavior(TransactionScopeRequired=true,
TransactionAutoComplete=false)]
    void TxOp1(string value) {... };

    [OperationContract(IsTerminating=true)]
    [TransactionFlow(TransactionFlowOption.Mandatory)]
    [OperationBehavior(TransactionScopeRequired=true,
TransactionAutoComplete=false)]
    void TxOp2(string value)
    {
        ...
        OperationContext.Current.SetTransactionComplete();
    }
}
```

The service and the clients that call the service use NetTcpBinding with transaction flow enabled. You need to configure the service so that when TxOp1 and TxOp2 are invoked under the same client session, they run under the same transaction context. What should you do?

A. Update the service contract to read as follows.

```
[ServiceContract(SessionMode=SessionMode.Required)]
```

Add the following behavior to the service implementation

```
[ServiceBehavior(InstanceContextMode=InstanceContextMode.PerSession)]
```

B. Update the service contract to read as follows.

```
[ServiceContract(SessionMode=SessionMode.Allowed)]
```

Add the following behavior to the service implementation.

```
[ServiceBehavior(InstanceContextMode=InstanceContextMode.Single,
ReleaseServiceInstanceOnTransactionComplete=false)]
```

C. Update the service contract to read as follows

```
[ServiceContract(SessionMode=SessionMode.Allowed)]
```

Add the following behavior to the service implementation.

```
[ServiceBehavior(InstanceContextMode=InstanceContextMode.Single)]
```

D. Update the service contract to read as follows.

```
[ServiceContract(SessionMode=SessionMode.Required)]
```

Add the following behavior to the service implementation.

```
[ServiceBehavior(InstanceContextMode=InstanceContextMode.PerCall)]
```

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

To participate in the same transaction both calls should be part of the same session. `SessionMode.Required` is required to enforce sessions for the endpoint.

InstanceContextMode Enumeration

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.instancecontextmode.aspx>)

SessionMode Enumeration

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.sessionmode.aspx>)

QUESTION 36

You are creating a Windows Communication Foundation (WCF) service application. The application needs to service many clients and requests simultaneously. The application also needs to ensure subsequent individual client requests provide a stateful conversation. You need to configure the service to support these requirements. Which attribute should you add to the class that is implementing the service?

- A. `[ServiceBehavior(InstanceContextMode = InstanceContextMode.PerSession, ConcurrencyMode = ConcurrencyMode.Single)]`
- B. `[ServiceBehavior(InstanceContextMode = InstanceContextMode.PerCall, ConcurrencyMode = ConcurrencyMode.Reentrant)]`
- C. `[ServiceBehavior(InstanceContextMode = InstanceContextMode.PerSession, ConcurrencyMode = ConcurrencyMode.Multiple)]`
- D. `[ServiceBehavior(InstanceContextMode = InstanceContextMode.PerCall, ConcurrencyMode = ConcurrencyMode.Multiple)]`

Correct Answer: C

Section: (none)

Explanation**Explanation/Reference:**

many clients and requests simultaneously --> `ConcurrencyMode.Multiple`
requests provide a stateful conversation --> `InstanceContextMode.PerSession`

ConcurrencyMode Enumeration

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.concurrencymode.aspx>)

InstanceContextMode Enumeration

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.instancecontextmode.aspx>)

QUESTION 37

You are integrating a Windows Communication Foundation (WCF) service within an enterprise-wide Service Oriented Architecture (SOA).

Your service has the following service contract:

```
[ServiceContract]
public class CreditCardConfirmationService
{
    [OperationContract]
    boolean ConfirmCreditCard(string ccNumber);

    double OrderAmount(int orderNumber);
}
```

You need to allow the code in the `ConfirmCreditCard` method to participate automatically in existing transactions.

If there is no existing transaction, a new transaction must be created automatically. What should you do?

- A. Inside the `ConfirmCreditCard` method, surround the code that must participate in the transaction with a `using(new TransactionScope())` block.
- B. Inside the `ConfirmCreditCard` method, surround the code that must participate in the transaction with a `using(new CommittableTransaction())`

block.

- C. Add an [OperationBehavior(TransactionScopeRequired=true)] attribute to the ConfirmCreditCard method.
- D. Add an [OperationBehavior(TransactionAutoComplete=true)] attribute to the ConfirmCreditCard method.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

Set the **TransactionScopeRequired** to true to require your operation to execute within a transaction scope.

If a flowed transaction is available, the operation executes within that transaction. If one is not available, a new transaction is created and used for the operation execution. The binding specified in the endpoint controls whether flowed transactions are supported.

Therefore, to obtain the proper behavior you must understand the interaction between whether transaction flow is permitted

by the binding and the TransactionScopeRequired property. The following table shows the possible behavior.

How to: Create a Transactional Service

(<http://msdn.microsoft.com/en-us/library/ms730232.aspx>)

QUESTION 38

Your Windows Communication Foundation (WCF) client application uses HTTP to communicate with the service.

You need to enable message logging and include all security information such as tokens and nonces in logged messages.

What should you do?

- A. In the application configuration file, add the logKnownPii attribute to the message logging diagnostics source and set the value of the attribute to true.
Generate the ContosoService class using the Add Service Reference wizard.
Add a reference to System.ServiceModel.Routing.dll.

Add the following code segment:

```
ContosoService client = new ContosoService();  
SoapProcessingBehavior behavior = new SoapProcessingBehavior();  
behavior.ProcessMessages = true;  
client.Endpoint.Behaviors.Add(behavior);
```

- B. In the application configuration file, add the following XML segment to the system.serviceModel configuration section group.

```
<diagnostics>  
  <messageLogging LogMessagesAtTransportLevel="true"  
  LogEntireMessage="true" />  
</diagnostics>
```

- C. In the machine configuration file, add the following XML segment to the system.serviceModel configuration section.

```
<machineSettings enableLoggingKnownPii="true" />
```

Generate the ContosoService class using the Add Service Reference wizard. Add the following code segment.

```
ContosoService client = new ContosoService();  
client.Endpoint.Behaviors.Add(new CallbackDebugBehavior(true));
```

- D. In the machine configuration file, add the following XML segment to the system.serviceModel configuration section.

```
<machineSettings enableLoggingKnownPii="true" />
```

In the application configuration file, add the logKnownPii attribute to the message logging diagnostics source and set the value of the attribute to true.

In the application configuration file, add the following XML segment to the system.serviceModel configuration section group.

```
<diagnostics>
  <messageLogging LogMessagesAtTransportLevel="true" />
</diagnostics>
```

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

Security Concerns for Message Logging

(<http://msdn.microsoft.com/en-us/library/ms730318.aspx>)

By default, keys and personally identifiable information (PII) such as username and password are not logged

in traces and logged messages. A machine administrator, however, can use the enableLoggingKnownPii attribute

in the machineSettings element of the Machine.config file to permit applications running on the machine to log

known personally identifiable information (PII). The following configuration demonstrates how to do this:

```
<configuration>
  <system.serviceModel>
    <machineSettings enableLoggingKnownPii="true" />
  </system.serviceModel>
</configuration>
```

An application deployer can then use the logKnownPii attribute in either the App.config or Web.config file to enable PII logging as follows:

```
<system.diagnostics>
  <sources>
    <source name="System.ServiceModel.MessageLogging"
      logKnownPii="true">
      <listeners>
        <add name="messages"
          type="System.Diagnostics.XmlWriterTraceListener"
          initializeData="c:\logs\messages.svclog" />
      </listeners>
    </source>
  </sources>
</system.diagnostics>
```

Only when both settings are true is PII logging enabled.

The combination of two switches allows the flexibility to log known PII for each application.

PII Logging Configuration

Logging of PII is controlled by two switches: a computer-wide setting found in Machine.config that allows a computer administrator to permit or deny logging of PII and an application setting that allows an application

administrator to toggle logging of PII for each source in a Web.config or App.config file.

The computer-wide setting is controlled by setting enableLoggingKnownPii to true or false, in the machineSettings element in Machine.config. For example, the following allows applications to turn on logging of PII.

```
<configuration>
  <system.serviceModel>
    <machineSettings enableLoggingKnownPii="true" />
  </system.serviceModel>
```

```
</configuration>
```

Note:

The Machine.config file has a default location: %WINDIR%\Microsoft.NET\Framework\v2.0.50727\CONFIG.

If the enableLoggingKnownPii attribute is not present in Machine.config, logging of PII is not allowed.

Enabling logging of PII for an application is done by setting the logKnownPii attribute of the source element to true or false

in the Web.config or App.config file. For example, the following enables logging of PII for both message logging and trace logging.

```
<configuration>
  <system.diagnostics>
    <sources>
      <source name="System.ServiceModel.MessageLogging"
logKnownPii="true">
        <listeners>
          ...
        </listeners>
      </source>
      <source name="System.ServiceModel" switchValue="Verbose,
ActivityTracing">
        <listeners>
          ...
        </listeners>
      </source>
    </sources>
  </system.diagnostics>
</configuration>
```

QUESTION 39

A Windows Communication Foundation (WCF) service has the following contract.

```
[ServiceContract(Namespace="http://contoso.com")]
public interface IShipping
{
    [OperationContract]
    string DoWork(int id);
}
```

This is one of several service contracts hosted by your application. All endpoints use SOAP 1.2 bindings with WS-Addressing 1.0.

The System.ServiceModel.MessageLogging trace source in the system.diagnostics configuration section is configured with one listener.

You need to make sure that only the messages that are returned from the DoWork operation are logged.

Which XML segment should you add to the system.serviceModel/diagnostics/messageLogging/filters configuration element?

- A. <add xmlns:addr="http://www.w3.org/2005/08/addressing">addr:Action[text()='http://contoso.com/IShipping/DoWorkResponse'] </add>
- B. <add xmlns:soap="http://www.w3.org/2003/05/soap-envelope">soap:Action[text()='http://contoso.com/IShipping/DoWorkResponse'] </add>
- C. <add xmlns:addr="http://www.w3.org/2005/08/addressing">addr:Action[text()='http://contoso.com/IShipping/DoWork'] </add>
- D. <add xmlns:soap="http://www.w3.org/2003/05/soap-envelope">soap:Action[text()='http://contoso.com/IShipping/DoWork'] </add>

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 40

You are implementing a Windows Communication Foundation (WCF) service contract named `IContosoService` in a class named `ContosoService`.

The service occasionally fails due to an exception being thrown at the service.

You need to send the stack trace of any unhandled exceptions to clients as a fault message. What should you do?

- A. In the application configuration file on the client, add the following XML segment to the `system.serviceModel/behaviors` configuration section group.

```
<endpointBehaviors>
  <behavior name="debug">
    <callbackDebug includeExceptionDetailInFaults="true" />
  </behavior>
</endpointBehaviors>
```

Associate the debug behavior with any endpoints that need to return exception details.

- B. In the application configuration file on the service and all the clients, add the following XML segment to the `system.diagnostics/sources` configuration section group.

```
<source name="System.ServiceModel" switchValue="Error"
propagateActivity="true">
  <listeners>
    <add name="ServiceModelTraceListener"
initializeData="appntracelog.svclog"
type="System.Diagnostics.XmlWriterTraceListener" />
  </listeners>
</source>
```

- C. Apply the following attribute to the `ContosoService` class.

```
[ServiceBehavior(IncludeExceptionDetailInFaults = true)]
```

- D. For each `OperationContract` exposed by `IContosoService`, apply the following attribute.

```
[FaultContract(typeof(Exception))]
```

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

ServiceBehaviorAttribute.IncludeExceptionDetailInFaults Property

Gets or sets a value that specifies that general unhandled execution exceptions are to be converted into a `System.ServiceModel.FaultException<TDetail>` of type `System.ServiceModel.ExceptionDetail` and sent as a fault message.

Set this to true only during development to troubleshoot a service.

QUESTION 41

A Windows Communication Foundation (WCF) service is generating a separate namespace declaration for each body member of a message contract, even though all body members share the same namespace.

You need to simplify the XML representation of your message contract so that the namespace is only declared once. What should you do?

- A. Declare a wrapper namespace for the message contract by using the `WrapperNamespace` property of the `MessageContract` attribute
- B. Explicitly set the `Namespace` property of all the `MessageBodyMember` attributes to the same namespace.
- C. Declare all of the body members as properties of a `DataContract` class and use the class as the only body member of the message contract.
- D. Declare all of the body members as properties of a separate `MessageContract` class and use the class as the only body member of the message contract.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 42

You are developing a Windows Communication Foundation (WCF) service.

You write a method named Submit that accepts messages of the type `System.ServiceModel.Channels.Message`.

You need to process the body of the incoming messages multiple times in the method.

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

- A. Use the `GetBody` method of the `Message` class to read the content of the messages.
- B. Use the `CreateBufferedCopy` method of the `Message` class to load the messages into memory.
- C. Use the `WriteBodyContents` method of the `BodyWriter` class to make a copy of the messages.
- D. Use the `CreateMessage` method of the `MessageBuffer` class to make a copy of the messages.

Correct Answer: BD

Section: (none)

Explanation

Explanation/Reference:

The body of a `Message` instance can only be accessed or written once. If you want to access a `Message` instance more than once, you should use the `MessageBuffer` class to completely store an entire `Message` instance into memory. A `MessageBuffer` instance is constructed by calling `CreateBufferedCopy` of a `Message` instance.

Calling **`MessageBuffer.CreateMessage()`** method creates an identical copy of the original `Message` instance you previously provided to the `CreateBufferedCopy` method of a `Message` instance. You can then save the message to a durable storage.

Message.CreateBufferedCopy Method

([http://msdn.microsoft.com/en-us/library/system.servicemodel.channels.message.createbufferedcopy\(v=vs.95\).aspx](http://msdn.microsoft.com/en-us/library/system.servicemodel.channels.message.createbufferedcopy(v=vs.95).aspx))

MessageBuffer.CreateMessage Method

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.channels.messagebuffer.createmessage.aspx>)

Example:

```
private byte[] ConvertMessageToByteArray(ref Message message)
{
    MemoryStream stream = new MemoryStream();
    XmlWriterSettings settings = new XmlWriterSettings();
    settings.Encoding = System.Text.Encoding.UTF8;
    XmlWriter writer = XmlWriter.Create(stream, settings);

    MessageBuffer buffer = message.CreateBufferedCopy(int.MaxValue);
message = buffer.CreateMessage();

    message.WriteMessage(writer);
    message = buffer.CreateMessage();
    writer.Flush();
    stream.Flush();
    byte[] retval = stream.ToArray();
    return retval;
}
```

QUESTION 43

You are creating a Windows Communication Foundation (WCF) service that responds using plain-old XML

(POX).

You have the following requirements:

- You must enable the /catalog.svc Items operation to respond using the POX, JSON, or ATOM formats.
- You also must ensure that the same URL is used regardless of the result type.
- You must determine the response format by using the Accepts HTTP header.

What should you do?

- A. Implement the `IChannelInitializer` interface in the service class.
- B. Implement the `System.Runtime.Serialization.IFormatterConverter` interface in the service class.
- C. Set the `BodyStyle` parameter of the `WebGet` attribute on the operation to `WebMessageBodyStyle.WrappedResponse`.
- D. Set the return type of the operation to `System.ServiceModel.Channels.Message`. Use the current `WebOperationContext` methods to return the data in the required format.

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

WebOperationContext A helper class that provides easy access to contextual properties of Web requests and responses.

WebOperationContext Class

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.web.weboperationcontext.aspx>)

QUESTION 44

A Windows Communication Foundation (WCF) solution uses two services to manage a shopping cart. Service A processes messages containing line items that total between \$0 and \$500.

Service B processes messages containing line items that total more than \$500. All messages are of equal importance to the business logic.

You need to route incoming messages to the appropriate services by using WCF routing.

Which two message filters should you add to the router? (Each correct answer presents part of the solution. Choose two.)

- A. a message filter with a priority of 100 that will forward messages that total between \$0 and \$500 to Service A
- B. a message filter with a priority of 0 that will forward messages that total between \$0 and \$500 to Service A
- C. a message filter with a priority of 0 that will forward all messages to Service B
- D. a message filter with a priority of 100 that will forward all messages to Service B

Correct Answer: AC

Section: (none)

Explanation

Explanation/Reference:

Message Filters

To implement content-based routing, the Routing Service uses `MessageFilter` implementations that inspect specific sections of a message, such as the address, endpoint name, or a specific XPath statement.

If none of the message filters provided with .NET Framework 4 meet your needs, you can create a custom filter by creating a new implementation of the base `MessageFilter` class.

When configuring the Routing Service, you must define filter elements (`FilterElement` objects) that describe the type of `MessageFilter` and any supporting data required to create the filter, such as specific string values

to search for within the message. Note that creating the filter elements only defines the individual message filters;
to use the filters to evaluate and route messages you must also define a filter table (FilterTableEntryCollection).

Each entry in the filter table references a filter element and specifies the client endpoint that a message will be routed to if the message matches the filter. The filter table entries also allow you to specify a collection of backup endpoints (BackupEndpointCollection), which defines a list of endpoints that the message will be transmitted to in the event of a transmission failure when sending to the primary endpoint. These endpoints will be tried in the order specified until one succeeds.

Priority is applied from high to low.

Routing Service

(<http://msdn.microsoft.com/en-us/library/ee517423.aspx>)

Routing Introduction

(<http://msdn.microsoft.com/en-us/library/ee517422.aspx>)

Message Filters

(<http://msdn.microsoft.com/en-us/library/ee517424.aspx>)

EXAMPLE:

```
<behaviors>
  <serviceBehaviors>
    <behavior name="routingData">
      <serviceMetadata httpGetEnabled="True"/>
      <!-- Add the RoutingBehavior and specify the Routing Table to use -->
      <routing filterTableName="routingTable1" />
    </behavior>
  </serviceBehaviors>
</behaviors>
<!--ROUTING SECTION -->
<routing>
  <filters>
    <filter name="MatchAllFilter1" filterType="MatchAll" />
    <filter name="RoundingFilter1" filterType="EndpointAddress"
      filterData="http://localhost:8000/routingservice/router/rounding" /
  >
</filters>
  <filterTables>
    <table name="routingTable1">
      <filters>
        <add filterName="MatchAllFilter1" endpointName="CalculatorService" />
        <add filterName="RoundingFilter1" endpointName="RoundingCalcService" />
      </filters>
    </table>
  </filterTables>
</routing>
```

QUESTION 45

You have an existing Windows Communication Foundation (WCF) service. You need to ensure that other services are notified when the service is started. What should you do?



<http://www.gratisexam.com/>

- A. Add the following standard endpoint to the service.
`<endpoint name="udpAnnouncementEndpoint" kind="udpDiscoveryEndpoint" />`
- B. Add the following standard endpoint to the service.
`<endpoint name="udpDiscoveryEndpoint" kind="udpAnnouncementEndpoint" />`
- C. Add a service behavior with the following element.
`<serviceDiscovery>
 <announcementEndpoints>
 <endpoint kind="udpDiscoveryEndpoint" />
 </announcementEndpoints>
</serviceDiscovery>`
- D. Add a service behavior with the following element.
`<serviceDiscovery>
 <announcementEndpoints>
 <endpoint kind="udpAnnouncementEndpoint" />
 </announcementEndpoints>
</serviceDiscovery>`

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

A standard endpoint that is used by services to send announcement messages over a UDP binding. This endpoint inherits from AnnouncementEndpoint and similarly has a fixed contract and supports two discovery versions.

In addition it has a fixed UDP binding and a default address value as specified in the WS-Discovery specifications (WS-Discovery April 2005 or WS-Discovery version 1.1).

Therefore, in the simplest case you do not have to specify any of these values when you create an AnnouncementEndpoint and add it to a service.

eg:

```
<serviceDiscovery>
  <announcementEndpoints>
    <endpoint kind="udpAnnouncementEndpoint"/>
  </announcementEndpoints>
</serviceDiscovery>
```

Configuring Discovery in Configuration File

(<http://msdn.microsoft.com/en-us/library/dd456792.aspx>)

QUESTION 46

You are developing a Windows Communication Foundation (WCF) service that reads messages from a public non-transactional MSMQ queue.

You need to configure the service to read messages from the failed-delivery queue.

Which URI should you specify in the endpoint configuration settings of the service?

- A. `net.msmq://localhost/msmq$FailedMessages`
- B. `net.msmq://localhost/msmq$DeadLetter`
- C. `net.msmq://localhost/system$DeadXact`
- D. `net.msmq://localhost/system$DeadLetter`

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

To read messages from a system non-transactional dead-letter queue, the URI must be of the form: `net.msmq://localhost/system$DeadLetter`.

Using Dead-Letter Queues to Handle Message Transfer Failures (<http://msdn.microsoft.com/en-us/library/ms789035.aspx>)

QUESTION 47

You have an existing Windows Communication Foundation (WCF) service that exposes a service contract over HTTP.

You need to expose that contract over HTTP and TCP. What should you do?

- A. Add a net.tcp base address to the host.
- B. Add an endpoint configured with a netTcpBinding.
- C. Add an endpoint behavior named netTcpBehavior to the existing endpoint.
- D. Add a binding configuration to the existing endpoint named netTcpBinding.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

```
<system.serviceModel>
  <services>
    <service behaviorConfiguration="WcfServiceLibrary1.Service1Behavior"
name="WcfServiceLibrary1.Service1">
      <endpoint address="" binding="netTcpBinding" bindingConfiguration=""
contract="WcfServiceLibrary1.IService1">
        <identity>
          <dns value="localhost" />
        </identity>
      </endpoint>
      <endpoint address="mex" binding="mexTcpBinding" bindingConfiguration=""
contract="IMetadataExchange" />
    </service>
  </services>
  <host>
    <baseAddresses>
      <add baseAddress="net.tcp://localhost:8523/Service1" />
    </baseAddresses>
  </host>
</system.serviceModel>
```

QUESTION 48

You have an existing Windows Communication Foundation (WCF) Web service. The Web service is not responding to messages larger than 64 KB.

You need to ensure that the Web service can accept messages larger than 64 KB without generating errors. What should you do?

- A. Increase the value of maxReceivedMessageSize on the endpoint binding.
- B. Increase the value of maxRequestLength on the httpRuntime element.
- C. Increase the value of maxBufferSize on the endpoint binding.
- D. Increase the value of maxBufferPoolSize on the endpoint binding.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:**WSHttpBindingBase.MaxReceivedMessageSize Property**

Gets or sets the maximum size for a message that can be processed by the binding.

Type: System.Int64

The maximum size, in bytes, for a message that is processed by the binding. The default value is 65,536 bytes.

QUESTION 49

A Windows Communication Foundation (WCF) service is responsible for transmitting XML documents between systems.

The service has the following requirements:

- It must minimize the transmission size by attaching the XML document as is without using escape characters or base64 encoding.
- It must interoperate with systems that use SOAP but are not built on the .NET platform.

You need to configure the service to support these requirements. Which message encoding should you use?

- A. Binary message encoding
- B. MTOM (Message Transmission Optimization Mechanism) message encoding
- C. Text message encoding with message version set to none
- D. Text message encoding with message version set to SOAP 1.2

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

MTOM

SOAP Message Transmission Optimization Mechanism

MTOM describes a mechanism for optimizing the transmission and/or wire format of a SOAP message by selectively re-encoding portions of the message while still presenting an XML Infoset to the SOAP application.

MTOM also describes an Inclusion Mechanism that operates in a binding-independent way, plus a specific binding for HTTP.

QUESTION 50

You are moving a Windows Communication Foundation (WCF) service into production. You need to be able to monitor the health of the service.

You only want to enable all performance counter instances exposed by the ServiceModelService counter group.

Which element should you add to the system.serviceModel section in the application configuration file?

- A. `<diagnostics performanceCounters="ServiceOnly" />`
- B. `<diagnostics wmiProviderEnabled="true" performanceCounters="Off" />`
- C. `<diagnostics performanceCounters="All" />`
- D. `<diagnostics wmiProviderEnabled="true" />`

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

You can enable performance counters for a WCF service through the app.config configuration file of the WCF service as follows:

```
<configuration>
  <system.serviceModel>
    <diagnostics performanceCounters="All" />
  </system.serviceModel>
```

</configuration>

The performanceCounters attribute can be set to enable a specific type of performance counters. Valid values are

All All category counters (ServiceModelService, ServiceModelEndpoint and ServiceModelOperation) are enabled.

ServiceOnly Only ServiceModelService category counters are enabled.

Off ServiceModel* performance counters are disabled. This is the default value.

Performance Counters

(<http://msdn.microsoft.com/en-us/library/ms735098.aspx>)

QUESTION 51

Four Windows Communication Foundation (WCF) services are hosted in Microsoft Internet Information Services (IIS).

No behavior configuration exists in the web.config file. You need to configure the application so that every service and endpoint limits the number of concurrent calls to 50 and the number of concurrent sessions to 25.

Which XML segment should you add to the system.serviceModel configuration section of the web.config file?

- A. <behaviors>
 <serviceBehaviors>
 <behavior name="*">
 <serviceThrottling maxConcurrentCalls="50"
maxConcurrentSessions="25" />
 </behavior>
 </serviceBehaviors>
</behaviors>
- B. <behaviors>
 <serviceBehaviors>
 <behavior name="default">
 <serviceThrottling maxConcurrentCalls="50"
maxConcurrentSessions="25" />
 </behavior>
 </serviceBehaviors>
</behaviors>
- C. <behaviors>
 <serviceBehaviors>
 <behavior name="">
 <serviceThrottling maxConcurrentCalls="50"
maxConcurrentSessions="25" />
 </behavior>
 </serviceBehaviors>
</behaviors>
- D. <behaviors>
 <serviceBehaviors>
 <behavior name="ALL">
 <serviceThrottling maxConncurentCalls="50"
maxConcurrentSessions="25" />
 </behavior>
 </serviceBehaviors>
</behaviors>

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

According to Darth, name="" behaves the same as without specifying it at all.

The <behavior> Element

Each behavior element is identified by a name attribute and provides either a system-provided behavior, such as <throttling>, or a custom behavior. If no name is given then that behavior element corresponds to the default service or endpoint behavior.

<serviceThrottling>

Specifies the throttling mechanism of a Windows Communication Foundation (WCF) service.

maxConcurrentCalls

A positive integer that limits the number of messages that currently process across a ServiceHost. Calls in excess of the limit are queued. Setting this value to 0 is equivalent to setting it to Int32.MaxValue. The default is 16.

maxConcurrentInstances

A positive integer that limits the number of InstanceContext objects that execute at one time across a ServiceHost. Requests to create additional instances are queued and complete when a slot below the limit becomes available. The default is 16.

maxConcurrentSessions

A positive integer that limits the number of sessions a ServiceHost object can accept. The service will accept connections in excess of the limit, but only the channels below the limit are active (messages are read from the channel). Setting this value to 0 is equivalent to setting it to Int32.MaxValue. The default is 10.

QUESTION 52

Windows Communication Foundation (WCF) service is self-hosted in a console application. The service implements the ITimeService service interface in the TimeService class. You need to configure the service endpoint for HTTP communication. How should you define the service and endpoint tags?

- A. Define the service tag as follows:

```
<service name="TimeService">
```

Define the endpoint tag as follows:

```
<endpoint kind="TimeService" address="http://localhost:8080/TimeService"
        binding="wsHttpBinding" contract="ITimeService"/>
```

- B. Define the service tag as follows:

```
<service name="TimeService">
```

Define the endpoint tag as follows:

```
<endpoint kind="ITimeService" address="http://localhost:8080/TimeService"
        binding="wsHttpBinding" contract="ITimeService" />
```

- C. Define the service tag as follows:

```
<service name="ITimeService">
```

Define the endpoint tag as follows:

```
<endpoint name="TimeService" address="http://localhost:8080/TimeService"
        binding="wsHttpBinding" contract="ITimeService"/>
```

- D. Define the service tag as follows:

```
<service name="TimeService">
```

Define the endpoint tag as follows:

```
<endpoint address="http://localhost:8080/TimeService"
        binding="wsHttpBinding" contract="ITimeService"/>
```

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

Specifying an Endpoint Address

(<http://msdn.microsoft.com/en-us/library/ms733749.aspx>)

QUESTION 53

Windows Communication Foundation (WCF) service will be hosted in Microsoft Internet Information Services (IIS).

You create a new application in IIS to host this service and copy the service DLL to the bin directory of the application.

You need to complete the deployment of this service to IIS. What should you do next?

- A. Create an asmx file and add a @ServiceHost directive to this file.
Copy the file to the root of the application directory.
- B. Create an .asmx file and add a @Register directive to this file.
Copy the file to the bin directory of the application.
- C. Create a svc file and add a @ServiceHost directive to this file.
Copy the file to the root of the application directory.
- D. Create a .svc file and add a @Register directive to this file.
Copy the file to the bin directory of the application.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

How to: Host a WCF Service in IIS

(<http://msdn.microsoft.com/en-us/library/ms733766.aspx>)

"bin" deploy (preferred option)

1. compile your WCF service into a DLL (class library)
2. create a website in IIS6
3. copy the WCF DLL's into the website's .\bin folder
4. create a *.svc file in that website
5. add an appropriate web.config in the website folder to define your endpoints and service configuration etc.

Your WCF service will now be reachable at the website's base address, plus the name of the *.svc file, e.g. <http://myserver/someweb/MyService.svc>

Your *.svc would look something like this:

```
<%@ ServiceHost Language="C#" Debug="true"
Service="WCF_Simple_Service.HelloIndigoService" %>
```

The Service= attributes denotes the class implementing the service - fully qualified with its namespace.

QUESTION 54

You are developing a Windows Communication Foundation (WCF) service that will be hosted in Microsoft Internet Information Services (IIS) 7.0.

The service must be hosted in an IIS application named Info. You need to enable this service to be hosted in IIS by changing the web.config file.

Which XML segment should you add to the web.config file?

- A.

```
<serviceHostingEnvironment>
  <serviceActivations>
    <add relativeAddress="Info.svc" service="Info" />
  </serviceActivations>
```

```
</serviceHostingEnvironment>
```

- B.

```
<serviceHostingEnvironment>
  <serviceActivations>
    <add relativeAddress="Info" service="Info.svc" />
  </serviceActivations>
</serviceHostingEnvironment>
```
- C.

```
<serviceHostingEnvironment>
  <transportConfigurationTypes>
    <add name="Info" transportConfigurationType="Info.svc" />
  </transportConfigurationTypes>
</serviceHostingEnvironment>
```
- D.

```
<serviceHostingEnvironment>
  <transportConfigurationTypes>
    <add name="Info.svc" transportConfigurationType="FileNotRequired" /
  >
  </transportConfigurationTypes>
</serviceHostingEnvironment>
```

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

```
<serviceHostingEnvironment>
  <serviceActivations>
    <add relativeAddress="MyServiceAddress" service="Service"
factory="MyServiceHostFactory" />
  </serviceActivations>
</serviceHostingEnvironment>
```

When using configuration-based activation, inline code in .svc files is not supported.

The relativeAddress attribute must be set to a relative address such as "<sub-directory>/service.svc" or "~/<sub-directory>/service.svc".

A configuration exception is thrown if you register a relative address that does not have a known extension associated with WCF.

The relative address specified is relative to the root of the virtual application.

Due to the hierarchical model of configuration, the registered relative addresses at machine and site level are inherited by virtual applications.

Registrations in a configuration file take precedence over settings in a .svc, .xamlx, .xoml, or other file.

Any '\' (backslashes) in a URI sent to IIS/WAS are automatically converted to a '/' (forward slash).

If a relative address is added that contains a '\' and you send IIS a URI that uses the relative address, the backslash is converted to a forward

slash and IIS cannot match it to the relative address. IIS sends out trace information that indicates that there are no matches found.

Configuration-Based Activation

(<http://msdn.microsoft.com/en-us/library/dd807499.aspx>)

QUESTION 55

You are creating a Windows Communication Foundation (WCF) service.

You need to ensure that the service is compatible with ASP.NET to make use of the session state. Which binding should you use?

- A. NetTcpContextBinding
- B. BasicHttpContextBinding
- C. NetTcpBinding
- D. NetMsmqBinding

Correct Answer: B

Section: (none)**Explanation****Explanation/Reference:**

Argument for BasicHttpContextBinding
(<http://msdn.microsoft.com/en-us/library/ff650724.aspx>)

Use basicHttpBinding with transport security to ensure that the service is compatible with legacy ASMX clients.

How to: Configure WCF Service to Interoperate with ASP.NET Web Service Clients

(<http://msdn.microsoft.com/en-us/library/ms731134.aspx>)

<basicHttpContextBinding>

(<http://msdn.microsoft.com/en-us/library/bb924439.aspx>)

BasicHttpContextBinding <basicHttpContextBinding>

A binding that is suitable for communicating with WS-Basic Profile conformant Web services that enables HTTP cookies to be used to exchange context.

Services **running in ASP.NET Compatibility mode** participate fully in the ASP.NET application pipeline and can make use

of ASP.NET features such as file/URL authorization, session state, and the HttpContext class.

The HttpContext class allows access to cookies, sessions, and other ASP.NET features.

This mode requires that the bindings use the HTTP transport and the service itself must be hosted in IIS.

ASP.NET Compatibility

(<http://msdn.microsoft.com/en-us/library/ms752234.aspx>)

QUESTION 56

A Windows Communication Foundation (WCF) client communicates with a service.

You created the client proxy by using Add Service Reference in MS Visual Studio.

You need to ensure that the client accepts responses of up to 5 MB in size. What should you change in the configuration file'?

- A. the value of the maxBufferPoolSize attribute to 5242880
- B. the value of the maxReceivedMessageSize attribute to 5242880
- C. the value of the maxBytesPerRead attribute to 5242880
- D. the value of the maxBufferSize attribute to 5242880

Correct Answer: B

Section: (none)

Explanation**Explanation/Reference:****WSHttpBindingBase.MaxReceivedMessageSize Property**

Gets or sets the maximum size for a message that can be processed by the binding.

Type: System.Int64

The maximum size, in bytes, for a message that is processed by the binding. The default value is 65,536 bytes.

QUESTION 57

An ASP.NET application hosts a RESTful Windows Communication Foundation (WCF) service at / Services/Contoso.svc.

The service provides a JavaScript resource to clients. You have an explicit reference to the JavaScript in your page markup as follows.

```
<script type="text/javascript" src="/Services/Contoso.svc/js" />
```

You need to retrieve the debug version of the service JavaScript. What should you do?

- A. In the <%@ ServiceHost %> header for /Services/Contoso.svc, set the Debug attribute to true.
- B. In the <%@ Page %> header, set the Debug attribute to true.
- C. In the script tag, add a debug attribute and set its value to true.
- D. In the script tag, append debug to the src attribute.

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 58

You are implementing a Windows Communication Foundation (WCF) client application that consumes the ICatalog and ICatalog2 service interfaces.

You need to ensure that the client discovers services implementing these interfaces.

The services may already be online or may come online within a 30 second time limit. How should you use WCF Discovery to accomplish this?

- A. Create one FindCriteria object for each interface and set the Duration of each FindCriteria to 30 seconds.
Call the FindAsync method of the DiscoveryClient class twice, one time for each of the FindCriteria objects, to search for the services.
- B. Create one FindCriteria object for each interface and set the Duration of each FindCriteria to two seconds.
Create a loop that calls the Find method of the DiscoveryClient class to search for the services. Within each loop iteration, call the Find method of the DiscoveryClient class once for each of the FindCriteria objects.
Run the loop until a service is found or 30 seconds pass.
- C. Create a single FindCriteria object and add both interfaces to its ContractTypeNames collection. Set the criteria's Duration to two seconds.
Create a loop that calls the Find method of the DiscoveryClient class to search for the services. Within each loop iteration, call the Find method of the DiscoveryClient class to search for the services.
Run the loop until a service is found or 30 seconds pass.
- D. Create a single FindCriteria object and add both interfaces to the ContractTypeNames collection. Set the Duration to 30 seconds and use the FindAsync method of the DiscoveryClient class to search for the services.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

I saw the answer D!!

FindCriteria

FindCriteria has several properties, which can be grouped into search criteria, which specify what services you are looking for, and find termination criteria (how long the search should last). A FindCriteria can contain multiple search criteria.

By default, the service has to match all of the components otherwise it does not consider itself a matching service.

If you want to find services that only match some of the criteria, you can implement custom find logic on the service or you can use multiple queries.

Search criteria include:

ContractTypeNames - Optional. The contract name of the service being searched for and the criteria

typically used when searching for a service.

If more than one contract name is specified, only service endpoints matching ALL contracts reply.

Note that in WCF an endpoint can only support one contract.
Scopes - Optional. Scopes are absolute URIs that are used to categorize individual service endpoints.

You may want to use this in scenarios where multiple endpoints expose the same contract and you want a way to search for a subset of the endpoints. If more than one scope is specified, only service endpoints matching ALL scopes reply.

ScopeMatchBy - Specifies the matching algorithm to use while matching the scopes in the Probe message with that of the endpoint.

There are five supported scope-matching rules:

ScopeMatchByExact does a basic case-sensitive string comparison.

ScopeMatchByPrefix matches by segments separated by "/". A search for `http://contoso/building1` matches a service with scope `http://contoso/building/floor1`.

Note that it does not match `http://contoso/building100` because the last two segments do not match.

ScopeMatchByLdap matches scopes by segments using an LDAP URL.

ScopeMatchByUuid matches scopes exactly using a UUID string.

ScopeMatchByNone matches only those services that do not specify a scope.

If a scope-matching rule is not specified, ScopeMatchByPrefix is used.

Termination criteria include:

Duration - The maximum time to wait for replies from services on the network. The default duration is 20 seconds.

MaxResults - The maximum number of replies to wait for. If MaxResults replies are received before Duration has elapsed, the find operation ends.

Discovery Find and FindCriteria

(<http://msdn.microsoft.com/en-us/library/ee816866.aspx>)

QUESTION 59

A Windows Communication Foundation (WCF) service sends notifications when the service is started and stopped.

You need to implement a client that logs these notifications. Which class should you use?

- A. AnnouncementService
- B. AnnouncementClient
- C. DiscoveryClient
- D. HttpListener

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

AnnouncementService Class

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.discovery.announcementsservice.aspx>)

AnnouncementEndpoint Class

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.discovery.announcementendpoint.aspx>)

AnnouncementService

This class listens for incoming messages on a standard announcement endpoint (AnnouncementEndpoint) and provides event notification when Hello or Bye announcement messages arrive.

You can provide a custom announcement service implementation by deriving from this class.

QUESTION 60

You are using Windows Communication Foundation (WCF) to create a service.

You need to implement a custom message-level security binding element. Which binding element should you use?

- A. TransportSecurityBindingElement
- B. HttpsTransportBindingElement
- C. SslStreamSecurityBindingElement
- D. WindowsStreamSecurityBindingElement

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

Three binding elements relate to **message level security**, all of which derive from the SecurityBindingElement class.

The three are **TransportSecurityBindingElement**, **SymmetricSecurityBindingElement**, and **AsymmetricSecurityBindingElement**.

The TransportSecurityBindingElement is used to provide Mixed mode security. The other two elements are used when the message layer provides security.

Additional classes are used when **transport level security** is provided:

- HttpsTransportBindingElement
- SslStreamSecurityBindingElement
- WindowsStreamSecurityBindingElement

How to: Create a Custom Binding Using the SecurityBindingElement

(<http://msdn.microsoft.com/en-us/library/ms730305.aspx>)

QUESTION 61

The endpoint of a Windows Communication Foundation (WCF) service uses basicHttpBinding for its binding.

Your company's policies have changed to require that messages not be sent in clear text.

You must ensure that all messages are encrypted when traveling across the network. What should you do?

- A. Set the ProtectionLevelAttribute on the service contract and update the binding attribute in the endpoint element of the configuration file to wsHttpBinding.
- B. Set the ProtectionLevelAttribute on the service contract and update the bindingConfiguration attribute in the endpoint element of the configuration file to webHttpBinding.
- C. Set the PrincipalPermissionAttribute on the service contract and update the binding attribute in the endpoint element of the configuration file to wsHttpBinding.
- D. Set the PrincipalPermissionAttribute on the service contract and update the bindingConfiguration attribute in the endpoint element of the configuration file to wsHttpBinding.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

ProtectionLevel Property

You can set the protection level by applying an appropriate attribute and setting the property.

You can set protection at the service level to affect all parts of every message, or you can set protection at increasingly granular levels, from methods to message parts.

By default, **basicHttpBinding** has security disabled.

wsHttpBinding Defines a secure, reliable, interoperable binding suitable for non-duplex service contracts.

The binding implements the following specifications: WS-Reliable Messaging for reliability, and WS-Security for

message security and authentication. The transport is HTTP, and message encoding is text/XML encoding.

By default, it provides message security using Windows authentication.

How to: Set the ProtectionLevel Property

(<http://msdn.microsoft.com/en-us/library/aa347791.aspx>)

QUESTION 62

You are creating a Windows Communication Foundation (WCF) service based on WSHttpBinding. New audit requirements dictate that callers must be authenticated on every call to ensure that their credentials have not been revoked.

You need to ensure that the service will not cache the security request token. What should you do?

- A. Apply a ServiceBehavior attribute to the service implementation class with the InstanceContextMode property set to Single.
- B. In the message security configuration, change clientCredentialType from IssuedToken to UserName.
- C. In the message security configuration, set establishSecurityContext to false.
- D. At the end of every operation, call the SessionStateUtility.RaiseSessionEnd method.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

SecurityPolicyAssertion.EstablishSecurityContext Property

Gets or sets a value indicating whether a secure conversation is established using SecurityContextToken security tokens.

SecurityContextToken Class Represents a security context token, which is used for signing and/or encrypting SOAP messages.

The **RaiseSessionEnd()** method is used by a session-state module to execute the Session_OnEnd event defined in the Global.asax file for an ASP.NET application. A session-state module will call the RaiseSessionEnd method when a session has been abandoned, or if the session expires.

SecurityPolicyAssertion.EstablishSecurityContext Property

(<http://msdn.microsoft.com/en-us/library/microsoft.web.services3.design.securitypolicyassertion.establishsecuritycontext.aspx>)

QUESTION 63

You have a self-hosted Windows Communication Foundation (WCF) service.

You need to configure the service to provide an X509 certificate during authentication. What should you use to configure the service?

- A. the Certificate property of the X509CertificateInitiatorServiceCredential class
- B. the SetCertificate method of the X509CertificateInitiatorServiceCredential class
- C. the SetCertificate method of the X509CertificateRecipientServiceCredential class
- D. the TrustedStoreLocation property of the X509CertificateRecipientServiceCredential class

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

X509CertificateRecipientServiceCredential Defines a certificate used by a service to identify itself.

X509CertificateInitiatorServiceCredential Represents the settings used by the service to validate the certificate

presented by the clients. It also contains a certificate for the service to use for encrypting

responses or callbacks for clients when MutualCertificateDuplex message security authentication mode is used.

X509CertificateInitiatorClientCredential Defines a certificate used by a client to identify itself.

X509CertificateRecipientServiceCredential.SetCertificate(String)

Specifies the certificate to use for representing the service by specifying the subject distinguished name.

QUESTION 64

You are creating an ASP.NET web application that hosts several Windows Communication Foundation (WCF) services.

The services have ASP.NET Compatibility Mode enabled. Users authenticate with the Web application by using a cookie-based ASP.NET Forms Authentication model.

You add a service file named Authentication.svc that contains the following code segment:

```
<%@ ServiceHost Service="System.Web.ApplicationServices.AuthenticationService"
    Factory="System.Web.ApplicationServices.ApplicationServicesHostFactory"
    %>
```

You need to ensure that users can access the WCF services without having to re-authenticate.

Which two configuration settings should you add? (Each is part of a complete solution. Choose two.)

- A. In the system.web.Extensions.scripting.webServices.authenticationService element, set the enabled attribute to true.
- B. In the system.web.Extensions.scripting.webServices.profileService element, set the enabled attribute to true.
- C. Add a service endpoint with basicHttpBinding for the contract System.Web.ApplicationServices.AuthenticationService.
- D. Add a custom service behavior named AuthenticationService Type Behaviors with a service AuthenticationManager element that has service AuthenticationManagerType set to System.Web.Security.SQL Membership Provider

Correct Answer: AC

Section: (none)

Explanation

Explanation/Reference:

AuthenticationService Class Enables access to ASP.NET forms authentication as a Web service.

The AuthenticationService object enables you to authenticate users through a Windows Communication Foundation (WCF) service.

You use the WCF authentication service when you must authenticate users through ASP.NET membership from an application

that is outside the Web application that stores the user credentials.

The application must be able to send and consume message in the SOAP format. Through the

AuthenticationService class, you can log users in,

log users out, validate credentials, check authentication status, customize authentication, and set the authentication cookie.

The AuthenticationService class contains four methods that you should access only through a WCF service: the IsLoggedIn,

Login, Logout, and ValidateUser methods. To call these methods, you enable the authentication service on a Web server

and then connect a WCF-compatible client application to the Web service.

For information about how to configure the authentication service, see [How to: Enable the WCF Authentication Service](#).

To log users on, you pass the user credentials to the Login method. If the credentials are valid, the AuthenticationService class creates an authentication cookie.

If the authentication cookie has not expired, you know that the user's credentials have been authenticated and you do not have to validate the credentials again.

(Cookie-less authentication is not available through the AuthenticationService class.)

The AuthenticationService can raise two events: Authenticating and CreatingCookie.

The Authenticating event occurs when the user credentials are being validated.

Create an event handler for the Authenticating event to customize how user credentials are validated.

The CreatingCookie event occurs when the authentication cookie is being set after user credentials have

been validated.

Create an event handler for the CreatingCookie event to customize the authentication cookie.

The ValidateUser method checks user credentials for authentication, but it does not return an authentication ticket.

Use ValidateUser when a user has previously logged in and you must check that the credentials are still valid at the start of a new application session.

```
<system.web.extensions>
  <scripting>
    <webServices>
      <authenticationService enabled="true" requireSSL = "true"/>
    </webServices>
  </scripting>
</system.web.extensions>
<system.serviceModel>
  <services>
    <service name="System.Web.ApplicationServices.AuthenticationService"
      behaviorConfiguration="AuthenticationServiceTypeBehaviors">
      <endpoint contract="System.Web.ApplicationServices.AuthenticationService"
        binding="basicHttpBinding"
        bindingConfiguration="userHttps"
        bindingNamespace="http://asp.net/ApplicationServices/
v200"/>
      </service>
    </services>
  <bindings>
    <basicHttpBinding>
      <binding name="userHttps">
        <security mode="Transport" />
      </binding>
    </basicHttpBinding>
  </bindings>
  <behaviors>
    <serviceBehaviors>
      <behavior name="AuthenticationServiceTypeBehaviors">
        <serviceMetadata httpGetEnabled="true"/>
      </behavior>
    </serviceBehaviors>
  </behaviors>
  <serviceHostingEnvironment aspNetCompatibilityEnabled="true"/>
</system.serviceModel>
```

Services **running in ASP.NET Compatibility mode** participate fully in the ASP.NET application pipeline and can make use

of ASP.NET features such as file/URL authorization, session state, and the HttpContext class.

The HttpContext class allows access to cookies, sessions, and other ASP.NET features.

This mode requires that the bindings use the HTTP transport and the service itself must be hosted in IIS.

ASP.NET Compatibility

(<http://msdn.microsoft.com/en-us/library/ms752234.aspx>)

QUESTION 65

A self-hosted Windows Communication Foundation (WCF) service uses a secure HTTP binding with a custom principal permission mode.

The binding requires users to provide their Windows logon credentials. You need to retrieve the identity of the caller.

What are two possible properties you can use to achieve this goal? (Each correct answer presents a complete solution Choose two)

- A. Thread.CurrentPrincipal.Identity.Name
- B. HttpContext.Current.User.Identity.Name
- C. ServiceSecurityContext.Current.PrimaryIdentity.Name

D. `OperationContext.Current.ServiceSecurityContext.PrimaryIdentity.Name`

Correct Answer: CD

Section: (none)

Explanation

Explanation/Reference:

ServiceSecurityContext.PrimaryIdentity Gets the primary identity associated with the current setting.

The primary identity is obtained from the credentials used to authenticate the current user.

If the credential is an X.509 certificate, the identity is a concatenation of the subject name and the thumbprint (in that order).

The subject name is separated from the thumbprint with a semicolon and a space.

If the subject field of the certificate is null, the primary identity includes just a semicolon, a space, and the thumbprint.

eg:

```
[OperationContract]
public int GetAccountBalance()
{
    //Block unauthorized users. SecurityException will return the correct
    SOAP Fault for this situation.
    if (!OperationContext.Current.ServiceSecurityContext.WindowsIdentity.
    IsAuthenticated)
        throw new SecurityException();

    //Retrieve the data for the current user.
    return Database.GetBalanceForUser
    (OperationContext.Current.ServiceSecurityContext.WindowsIdentity.Name);
}
```

QUESTION 66

You are creating a Windows Communication Foundation (WCF) service that accepts claims-based tokens. You need to ensure that the service can use claims from trading partners even though there are variations on naming for the same elements.

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

- A. Register a custom Service Authorization Manager that implements Check Access. In this method, use `System.Convert.ChangeType` to transform the incoming claim set to a `WindowsClaimSet` type.
- B. Apply a `PrincipalPermission` attribute on the operation with the required claims listed in the `Roles` property.
- C. Within the operation, verify the presence of the required claims in the current `AuthorizationContext`.
- D. Register an `AuthorizationPolicy` that maps external claims to an internal `ClaimSet`.

Correct Answer: CD

Section: (none)

Explanation

Explanation/Reference:

Authorization Policy

(<http://msdn.microsoft.com/en-us/library/ms751416.aspx>)

Authorization policy

A set of rules for mapping a set of input claims to a set of output claims.

Evaluating authorization policy results in claim sets being added to an evaluation context and subsequently an authorization context.

QUESTION 67

A Windows Communication Foundation (WCF) service uses a list of application-defined roles for operations.

These roles are stored in a database. You need to authorize calls against the list of roles retrieved from the

database.

Which service behavior should you use to authorize the calls?

- A. <serviceAuthorization principalPermissionMode="None" roleProviderName="SqlProvider" />
- B. <serviceAuthorization principalPermissionMode="None" roleProviderName="SqlProvider" />
- C. <serviceAuthorization principalPermissionMode="None" roleProviderName="SqlProvider" />
- D. <serviceAuthorization principalPermissionMode="None" roleProviderName="SqlProvider" />

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

<serviceAuthorization> element .NET Framework 4

Specifies settings that authorize access to service operations

Syntax:

```
<system.serviceModel>
```

```
  <behaviors>
```

```
    <serviceBehaviors>
```

```
      <behavior>
```

```
        <serviceAuthorization>
```

```
<serviceAuthorization
```

```
  impersonateCallerForAllOperations="Boolean"
```

```
  principalPermissionMode="None/UseWindowsGroups/UseAspNetRoles/Custom"
```

```
  roleProviderName="String"
```

```
  serviceAuthorizationManagerType="String" />
```

```
  <authorizationPolicies>
```

```
    <add policyType="String" />
```

```
  </authorizationPolicies>
```

```
</serviceAuthorization>
```

Remarks

This section contains elements affecting authorization, custom role providers, and impersonation.

The **principalPermissionMode** attribute specifies the groups of users to use when authorizing use of a protected method.

The default value is **UseWindowsGroups** and specifies that Windows groups, such as "Administrators" or "Users,"

are searched for an identity trying to access a resource. You can also specify **UseAspNetRoles** to use a custom role provider

that is configured under the <system.web> element, as shown in the following code.

```
<system.web>
```

```
  <membership defaultProvider="SqlProvider" userIsOnlineTimeWindow="15">
```

```
    <providers>
```

```
      <clear />
```

```
      <add
```

```
        name="SqlProvider"
```

```
        type="System.Web.Security.SqlMembershipProvider"
```

```
        connectionStringName="SqlConn"
```

```
        applicationName="MembershipProvider"
```

```
        enablePasswordRetrieval="false"
```

```
        enablePasswordReset="false"
```

```
        requiresQuestionAndAnswer="false"
```

```
        requiresUniqueEmail="true"
```

```
        passwordFormat="Hashed" />
```

```
      </providers>
```

```
    </membership>
```

```
    <!-- Other configuration code not shown.-->
```

```
</system.web>
```

The following code shows the **roleProviderName** used with the **principalPermissionMode** attribute.

```
<behaviors>
  <behavior name="ServiceBehaviour">
    <serviceAuthorization principalPermissionMode ="UseAspNetRoles" roleProviderName ="SqlProvider" /
  >
  </behavior>
<!-- Other configuration code not shown. -->
</behaviors>
```

QUESTION 68

A Windows Communication Foundation (WCF) service is required to log all authorization attempts to the Windows Event Log.

You need to configure a behavior and apply it to the service to support this requirement. Which behavior should you configure and apply?

- A. serviceAuthenticationManager
- B. serviceAuthorization
- C. serviceCredentials
- D. serviceSecurityAudit

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

<serviceSecurityAudit> Specifies settings that enable auditing of security events during service operations.

<serviceCredentials> Specifies the credential to be used in authenticating the service and the client credential validation-related settings.

<serviceAuthorization> Specifies settings that authorize access to service operations

ServiceAuthenticationManager Class Represents a service authentication manager.

<serviceSecurityAudit>

(<http://msdn.microsoft.com/en-us/library/ms731694.aspx>)

QUESTION 69

A Windows Communication Foundation (WCF) service only accepts messages that are signed and encrypted.

A client application is not receiving expected responses from the service.

You need to enable logging to verify that the messages from the client are signed and encrypted.

You also need to see what each message looks like before the message body is deserialized into a NET object. What should you do?

- A. Configure the System.ServiceModel trace source in the system.diagnostics configuration section. In the system.serviceModel configuration, add the following XML segment:

```
<diagnostics>
  <messageLogging logEntireMessage="true" logMessagesAtServiceLevel="true"
logMessagesAtTransportLevel="true" />
</diagnostics>
```
- B. Configure the System.ServiceModel trace source in the system.diagnostics configuration section. In the system.serviceModel configuration, add the following XML segment:

```
<diagnostics>
  <messageLogging logEntireMessage="true" logMessagesAtServiceLevel="true"
/>
</diagnostics>
```
- C. Configure the System.ServiceModel.MessageLogging trace source in the system.diagnostics configuration section. In the system.serviceModel configuration, add the following XML segment:

```

<diagnostics>
  <messageLogging logEntireMessage="true" logMessagesAtServiceLevel="true"
logMessagesAtTransportLevel="true" />
</diagnostics>

```

D. Configure the System.ServiceModel.MessageLogging trace source in the system.diagnostics configuration section.

In the system.serviceModel configuration, add the following XML segment:

```

<diagnostics>
  <messageLogging logMessagesAtServiceLevel="true"
logMessagesAtTransportLevel="true" />
</diagnostics>

```

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

MessageLoggingElement properties:

LogEntireMessage Gets or sets a value that specifies whether the entire message (message header and body) is logged.

LogMessagesAtServiceLevel Gets or sets a value that specifies whether messages are traced at the service level (before encryption and transport-related transforms).

LogMessagesAtTransportLevel Gets or sets a value that specifies whether messages are traced at the transport level.

Configuring Message Logging

(<http://msdn.microsoft.com/en-us/library/ms730064.aspx>)

Example:

```

<system.diagnostics>
  <sources>
    <source name="System.ServiceModel.MessageLogging">
      <listeners>
        <add name="messages"
          type="System.Diagnostics.XmlWriterTraceListener"
          initializeData="c:\logs\messages.svclog" />
      </listeners>
    </source>
  </sources>
</system.diagnostics>

<system.serviceModel>
  <diagnostics>
    <messageLogging
      logEntireMessage="true"
      logMalformedMessages="false"
      logMessagesAtServiceLevel="true"
      logMessagesAtTransportLevel="false"
      maxMessagesToLog="3000"
      maxSizeOfMessageToLog="2000" />
  </diagnostics>
</system.serviceModel>

```

QUESTION 70

You want to debug the Windows Communication Foundation (WCF) client and server interaction through message and application tracing.

You need to correlate traces generated on the client and the server.

Which XML segment should you add to the system.diagnostics configuration element in the client and server application configuration file?

A. <sources>
 <source propagateActivity="true" name="System.ServiceModel"
 switchValue="Warning, ActivityTracing">

```
        <listeners>
            <add name="ServiceModelTraceListener" />
        </listeners>
    </source>
</sources>
```

B. <sources>
 <source name="System.ServiceModel.MessageLogging"
 switchValue="Verbose">
 <listeners>
 <add name="ServiceModelTraceListener" />
 </listeners>
 </source>
</sources>

C. <sources>
 <source name="System.ServiceModel.MessageLogging"
 propagateActivity="true" switchValue="Warning, ActivityTracing">
 <listeners>
 <add name="ServiceModelTraceListener" />
 </listeners>
 </source>
</sources>

D. <sources>
 <source name="System.ServiceModel"
 switchValue="VerboseActivityTracing">
 <listeners>
 <add name="ServiceModelTraceListener" />
 </listeners>
 </source>
</sources>

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

Configuring Activity Tracing and Propagation for Correlation

The activityTracing value specified for the switchValue attribute is used to enable activity tracing, which emits traces for activity boundaries and transfers within endpoints.

When you use certain extensibility features in WCF, you might get a NullReferenceException when activity tracing is enabled.

To fix this problem, check your application's configuration file and ensure that the switchValue attribute for your trace source is not set to activityTracing.

The **propagateActivity** attribute indicates whether the activity should be propagated to other endpoints that participate in the message exchange.

By setting this value to true, you can take trace files generated by any two endpoints and observe how a set of traces on one endpoint flowed to a set of traces on another endpoint.

Both propagateActivity and ActivityTracing Boolean values apply to the System.ServiceModel TraceSource.

The ActivityTracing value also applies to any trace source, including WCF or user-defined ones.

You cannot use the propagateActivity attribute with user-defined trace sources.

For user code activity ID propagation, make sure you do not set ServiceModel ActivityTracing, while still having ServiceModel propagateActivity attribute set to true.

Tracing and Message Logging

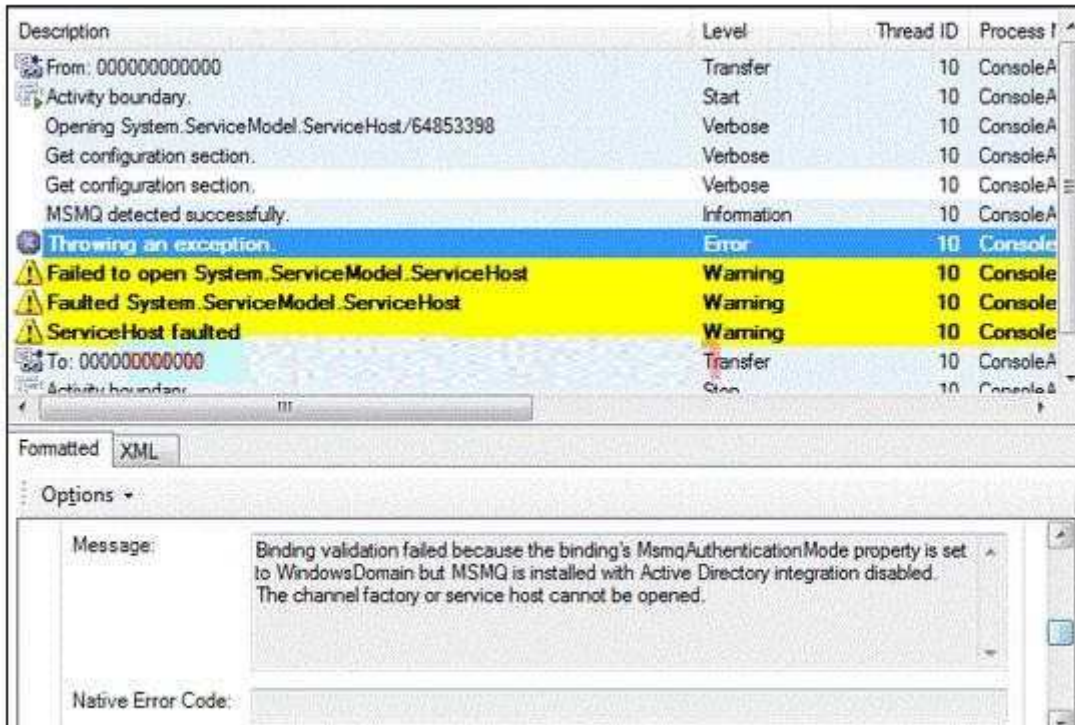
(<http://msdn.microsoft.com/en-us/library/ms751526.aspx>)

Configuring Tracing

(<http://msdn.microsoft.com/en-us/library/ms733025.aspx>)

QUESTION 71

You are using tracing to diagnose run-time issues. When you look at the traces for the service in SvcTraceViewer.exe, you see what is shown in the exhibit:



The exception trace is selected in SvcTraceViewer.exe. You need to interpret the trace results to determine where the error occurred and what to do next. What should you do?

- A. This issue occurred in the ServiceHost during ServiceHost.Open(). Enable WMI by adding the following configuration to the system.serviceModel configuration section in the application configuration file
`<diagnostics wmiProviderEnabled="true" />`
Restart the application and inspect the endpoints visible through WMI.
- B. This issue occurred in the ServiceHost during ServiceHost.Open(). Compare the security settings for any endpoints that use an MSMQ transport to the security configuration of the MSMQ queue used by the endpoint.
- C. This issue occurred at the ServiceHost when receiving a message. Compare the security configurations on the client and server to make sure that they are compatible.
- D. This issue occurred at the ServiceHost when accepting an initial set of messages from MSMQ. Log all messages sent between the client and the sever.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 72

You are adding a Windows Communication Foundation (WCF) service to an existing application. The application is configured as follows. (Line numbers are included for reference only)

```
01 <configuration>
02   <system.serviceModel>
03     <services>
04       <service name="Contoso.Sales.Stock.Service"
05         behaviorConfiguration="MetadataBehavior">
06         <host>
07           <baseAddresses>
08             <add baseAddress="http://contosso.com:8080/StockService" /
09           >
10             </baseAddresses>
11           </host>
12         </service>
13       </services>
14     <behaviors>
15       <serviceBehaviors>
16         <behavior name="MetadataBehavior">
17         </behavior>
18       </serviceBehaviors>
19     </behaviors>
20   </system.serviceModel>
21 </configuration>
```

You need to configure the service to publish the service metadata.

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

- A. Add the following XML segment between lines 10 and 11:
`<endpoint address="" binding="mexHttpBinding" contract="MetadataExchange" />`
- B. Add the following XML segment between lines 10 and 11:
`<endpoint address="" binding="basicHttpBinding" contract="MetadataExchange" />`
- C. Add the following XML segment between lines 15 and 16:
`<serviceDiscovery>
 <announcementEndpoints>
 <endpoint address="" />
 </announcementEndpoints>
</serviceDiscovery>`
- D. Add the following XML segment between lines 15 and 16:
`<serviceMetadata httpGetEnabled="true" />`

Correct Answer: AD

Section: (none)

Explanation

Explanation/Reference:

<mexHttpBinding>

Specifies the settings for a binding used for the WS-MetadataExchange (WS-MEX) message exchange over HTTP.

This binding is essentially a WSHttpBinding binding with security disabled. It supports most metadata requests.

<serviceMetadata>

Specifies the publication of service metadata and associated information.

httpGetEnabled

A Boolean value that specifies whether to publish service metadata for retrieval using an HTTP/Get request. The default is false.

<serviceMetadata>

(<http://msdn.microsoft.com/en-us/library/ms731317.aspx>)

<mexHttpBinding>

(<http://msdn.microsoft.com/en-us/library/aa967390.aspx>)

QUESTION 73

You are modifying a Windows Communication Foundation (WCF) service that issues security tokens. The service is accessible through the named pipe protocol. No endpoints are added in the service code. The configuration file for the service is as follows. (Line numbers are included for reference only)

```
01 <configuration>
02   <system.serviceModel>
03     <services>
04       <service name="Contoso.TokenService">
05         ...
06         <host>
07           <baseAddress>
08             ...
09             <add baseAddress="net.pipe://www.contoso.com/tokenpipe" />
10           <baseAddress>
11         </host>
12       </service>
13     </services>
14   </system.serviceModel>
15 </configuration>
```

You need to ensure that the existing client applications can access the service through HTTP and named pipes.

What should you do?

A. Insert the following at line 05:

```
<endPoint address="http://www.contoso.com" binding="wsHttpBinding"
contract="Contoso.TokenService" />
```

B. Insert the following at line 05:

```
<endPoint address="http://www.contoso.com" binding="basicHttpBinding"
contract="Contoso.TokenService" />
```

C. Insert the followin at line 08:

```
<add baseAddress="http://www.contoso.com" />
```

D. Insert the followin at line 08:

```
<add baseAddress="http://www.contoso.com:8080" />
```

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

You can also have multiple base addresses for a service, but each service is allowed only one base address for each transport.

If you have multiple endpoints, each of which is configured with a different binding, their addresses must be unique.

Endpoints that use the same binding but different contracts can use the same address.

Specifying an Endpoint Address

(<http://msdn.microsoft.com/en-us/library/ms733749.aspx>)

EXAMPLE:

if you have two base addresses, say:

```
http://localhost:9000/service
net.tcp://localhost:9001/service
```

and three endpoints

```
endpoint 1: address="LOB" binding="basicHttpBinding", contract = ...  
endpoint 2: address="LOB" binding="netTcpBinding", contract = ...  
endpoint 3: address="management" binding="wsHttpBinding", contract = ...
```

then they will be exposed at the following addresses irrespective of what order you declare them

```
endpoint 1: http://localhost:9000/service/LOB  
endpoint 2: net.tcp://localhost:9001/service/LOB  
endpoint 3: http://localhost:9000/service/management
```

QUESTION 74

You are hosting a Windows Communication Foundation (WCF) service under Microsoft Internet Information Services (IIS) 7.0.

You have set up a web site in IIS Manager. The physical path is c:\wwwroot\Calendar.

There is a Calendar.svc file in the c:\wwwroot\Calendar folder. It contains the following directive:

```
<% @ServiceHost Language="C#" Debug="true" Service="Calendar.Calendar"  
CodeBehind="CalendarSvc.cs" %>
```

The CalendarSvc.cs file contains the source for the Calendar class in the Calendar namespace.

You compile this code into the Calendar.dll file. You need to deploy your service to the web site. What should you do?

- A. Copy the Calendar.dll file to the c:\wwwroot\Calendar\code folder
- B. Copy the Calendar.dll file to the c:\wwwroot\Calendar\bin folder
- C. Copy the Calendar.svc.cs file to the c:\wwwroot\Calendar\bin folder
- D. Copy the Calendar.svc.cs file to the c:\wwwroot\Calendar\code folder

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

There are basically two options of WCF deployment in IIS:

Option 1 - "bin" deploy (preferred option)

1. compile your WCF service into a DLL (class library)
2. create a website in IIS6
3. copy the WCF DLL's into the website's .\bin folder
4. create a *.svc file in that website
5. add an appropriate web.config in the website folder to define your endpoints and service configuration etc.

Your WCF service will now be reachable at the website's base address, plus the name of the *.svc file, e.g. <http://myserver/someweb/MyService.svc>

Your *.svc would look something like this:

```
<%@ ServiceHost Language="C#" Debug="true"  
Service="WCF_Simple_Service.HelloIndigoService" %>
```

The Service= attributes denotes the class implementing the service - fully qualified with its namespace.

Option 2 - put stuff into App_Code

1. create a website in IIS6
2. put all your WCF related *.cs files directly into the .\App_Code folder
3. create a *.svc file in that website
4. add an appropriate web.config in the website folder to define your endpoints and service configuration etc.

Your WCF service will now be reachable at the website's base address, plus the name of the *.svc file, e.g. <http://myserver/someweb/MyService.svc>

Your *.svc would look something like this:


```
<%@ ServiceHost Language="C#" Debug="true" Service="Service"
CodeBehind="~/App_Code/Service.cs" %>
```

A simple, sample web.config might look something like this:

```
<system.serviceModel>
  <behaviors>
    <serviceBehaviors>
      <behavior name="WithDebug">
        <serviceMetadata httpGetEnabled="true" />
        <serviceDebug includeExceptionDetailInFaults="true" />
      </behavior>
    </serviceBehaviors>
  </behaviors>
  <serviceHostingEnvironment multipleSiteBindingsEnabled="true" />
<services>
  <service name="SimpleWCF.HelloIndigoService" behaviorConfiguration="true">
    <endpoint
      address=""
      binding="basicHttpBinding"
      contract="SimpleWCF.IHelloIndigoService" />
    <endpoint
      address="mex"
      binding="mexHttpBinding"
      contract="IMetadataExchange" />
  </service>
</services>
</system.serviceModel>
```

You basically define your <service> tag - and again: the name= denotes the class implementing the service - fully qualified with its namespace.

It must contain at least one endpoint - since IIS6 only support HTTP, you can use basicHttpBinding or wsHttpBinding and that's about all there is.

A "mex" endpoint is optional - but very useful, especially for development and testing.

It allows client to "discover" the service and get its service description so it can interface with it.

Once your service is deployed in IIS, you can see it in action using a tool like the WCF Test Client that ships for free with WCF, or SoapUI which is a general-purpose SOAP testing utility (with a free edition for you to use).

QUESTION 75

You are maintaining a Windows Communication Foundation (WCF) service that uses a custom username password class to authenticate clients with. The service certificate is hosted in the deployment server store for trusted root certificate authorities and has a Subject value of TaxServiceKey. Other service certificates hosted on the same server also use TaxServiceKey as a Subject value.

You need to ensure that the service identifies itself with a certificate whose subject name and distinguished names are TaxServiceKey.

Which code segment should you use?

- A.

```
HostInstance.Credentials.ServiceCertificate.SetCertificate(
    StoreLocation.LocalMachine, StoreName.My,
    X509FindType.FindBySubjectName,
    "CN=TaxServiceKey");
```
- B.

```
HostInstance.Credentials.ServiceCertificate.SetCertificate(
    StoreLocation.LocalMachine, StoreName.AuthRoot,
    X509FindType.FindBySubjectName,
    "CN=TaxServiceKey");
```
- C.

```
HostInstance.Credentials.ServiceCertificate.SetCertificate(
    StoreLocation.LocalMachine, StoreName.My,
    X509FindType.FindByDistinguishedName,
    "CN=TaxServiceKey");
```

```
D. HostInstance.Credentials.ServiceCertificate.SetCertificate(  
    StoreLocation.LocalMachine, StoreName.Root,  
    x509FindType.FindByDistinguishedName,  
    "CN="TaxServiceKey"");
```

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

StoreName Enumeration

Specifies the name of the X.509 certificate store to open.

StoreName Enumeration

(<http://msdn.microsoft.com/en-us/library/system.security.cryptography.x509certificates.storename.aspx>)

AddressBook	The X.509 certificate store for other users.
AuthRoot	The X.509 certificate store for third-party certificate authorities (CAs).
CertificateAuthority	The X.509 certificate store for intermediate certificate authorities (CAs).
Disallowed	The X.509 certificate store for revoked certificates.
My	The X.509 certificate store for personal certificates.
Root	The X.509 certificate store for trusted root certificate authorities (CAs).
TrustedPeople	The X.509 certificate store for directly trusted people and resources.
TrustedPublisher	The X.509 certificate store for directly trusted publishers.

X509FindType Enumeration

([http://msdn.microsoft.com/en-us/library/system.security.cryptography.x509certificates.x509findtype\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/system.security.cryptography.x509certificates.x509findtype(v=vs.85).aspx))

FindBySubjectName The findValue parameter for the Find method must be a string representing the subject name of the certificate.

This is a less specific search than if you use the FindBySubjectDistinguishedName enumeration value.

Using the FindBySubjectName value, the Find method performs a case-insensitive string compare using the supplied value.

For example, if you pass "MyCert" to the Find method, it will find all certificates with the subject name containing that string, regardless of other subject values. Searching by distinguished name is a more precise search.

FindBySubjectDistinguishedName The findValue parameter for the Find method must be a string representing the subject distinguished name of the certificate.

This is a more specific search than if you use the FindBySubjectName enumeration value.

Using the FindBySubjectDistinguishedName value, the Find method performs a case-insensitive string compare for the entire distinguished name. Searching by subject name is a less precise search.

QUESTION 76

You are hosting a Windows Communication Foundation (WCF) service at <http://www.contoso.com> for a law enforcement agency.

The agency adds operations to support sending biometric fingerprint data via non-buffered streaming.

The service data is routed between intermediaries. The WCF binding you are using by default does not support encryption.

You need to ensure that the fingerprint data is not disclosed when passed over the network. What should you do?

- A. Use basicHttpBinding with message security to <https://www.contoso.com>
- B. Use basicHttpBinding with transport security to <https://www.contoso.com>
- C. Use wsHttpBinding with message security to <https://www.contoso.com>
- D. Use wsHttpBinding with transport security to <http://www.contoso.com>

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 77

You create a Windows Communication Foundation (WCF) service and deploy it with wsHttpBinding and message security enabled.

You create an intermediate WCF service for logging messages sent to the primary service.

The intermediate service is called via endpoint behaviour. The primary service is receiving malformed data from a client application.

You need to enable inspection of the malformed data and prevent message tampering. What do you do?

- A. Specify a protection level of None in the contract for the intermediate service.
Disable message and transport security from the client application configuration file.
- B. Specify a protection level of Sign in the contract for the intermediate service.
Disable transport security from the client application configuration file.
- C. Modify the binding on the intermediate service to use netNamedPipeBinding.
- D. Modify the binding on the intermediate service to use webHttpBinding.

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

ProtectionLevel Indicates the security services requested for an authenticated stream.

Sign Sign data to help ensure the integrity of transmitted data.

None Authentication only.

<webHttpBinding>

(<http://msdn.microsoft.com/en-us/library/bb412176.aspx>)

<webHttpBinding> Defines a binding element that is used to configure endpoints for Windows Communication Foundation (WCF)

Web services that respond to HTTP requests instead of SOAP messages.

<netNamedPipeBinding>

(<http://msdn.microsoft.com/en-us/library/ms731291.aspx>)

<netNamedPipeBinding>

Defines a binding that is secure, reliable, optimized for on-machine cross process communication.

QUESTION 78

You develop a Windows Communication Foundation (WCF) service. You enable all performance counters and run multiple calls to the service.

The service must isolate session data to each user. You need to monitor the instancing behaviour used in the service.

Which performance counter should you monitor?

- A. ServiceModeService 4.0.0.0\Calls
- B. ServiceModeService 4.0.0.0\Instances
- C. ASP.Net StateService \ Server Sessions Active
- D. ASP.Net StateService \ Server Sessions Total

Correct Answer: B

Section: (none)**Explanation****Explanation/Reference:**

"monitor the instancing behaviour" --> ServiceModeService 4.0.0.0\Instances

QUESTION 79

You are developing a Windows Communication Foundation (WCF) service that allows customers to update financial data.

The service contract is defined as follows. (Line numbers are included for reference only)

```

01 [ServiceContract]
02 public interface IDataUpdate
03 {
04     [OperationContract]
05     [TransactionFlow(TransactionFlowOption.Mandatory)]
06     void Update(string accountNumber, double amount);
07 }
08
09 class UpdateService : IDataUpdate
10 {
11     [OperationBehavior(TransactionScopeRequired = true,
TransactionAutoComplete = false)]
12     public void Update(string accountNumber, double amount)
13     {
14         ...
15     }
16 }
17

```

You need to ensure that the service is invoked within the transaction. What should you do?

- A. Replace line 01 with the following code
`[ServiceContract(SessionMode = SessionMode.NotAllowed)]`
- B. Replace line 01 with the following code
`[ServiceContract(SessionMode = SessionMode.Required)]`
- C. Call the Invoke method of the form and supply a delegate.
- D. Call the BeginInvoke method of the form and supply a delegate.

Correct Answer: B

Section: (none)

Explanation**Explanation/Reference:**

The contract for the service defines that all of the operations require a transaction to be flowed with requests:

```

[ServiceContract(Namespace = "http://Microsoft.ServiceModel.Samples", SessionMode =
SessionMode.Required)]
public interface ICalculator
{
    [OperationContract]
    [TransactionFlow(TransactionFlowOption.Mandatory)]
    double Add(double n);
    [OperationContract]
    [TransactionFlow(TransactionFlowOption.Mandatory)]
    double Subtract(double n);
    [OperationContract]
    [TransactionFlow(TransactionFlowOption.Mandatory)]
    double Multiply(double n);
}

```

```
[OperationContract]
[TransactionFlow(TransactionFlowOption.Mandatory)]
double Divide(double n);
}
```

Service Transaction Behavior

(<http://msdn.microsoft.com/en-us/library/ms751413.aspx>)

QUESTION 80

You are developing a Windows Communication Foundation (WCF) service that is hosted by a Windows Forms Application.

The ServiceHost instance is created in the Form Constructor.

You need to ensure that the service is not blocked while the UI thread is busy. What should you do?

- A. Decorate the service implementation class with the following line of code
`[ServiceBehavior(UseSynchronizationContext = false)]`
- B. Decorate the service implementation class with the following line of code
`[ServiceBehavior(ConcurrencyMode = ConcurrencyMode.Multiple)]`
- C. Call the Invoke method of the form and supply a delegate.
- D. Call the BeginInvoke method of the form and supply a delegate.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

ServiceBehaviorAttribute.UseSynchronizationContext Property

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.servicebehaviorattribute.usesynchronizationcontext.aspx>)

ServiceBehaviorAttribute.UseSynchronizationContext Property

Gets or sets a value that specifies whether to use the current synchronization context to choose the thread of execution.

Use this property to provide the user interface thread affinity that some applications require. For example, a Windows Forms application may be registered as a singleton service object. In this case, all calls into the service must run on the Windows Forms thread. The default case, in which UseSynchronizationContext is set to true, synchronizes all calls to the service to run on the thread captured from the System.Threading.SynchronizationContext.

Writing Smart Clients by Using Windows Communication Foundation

(<http://msdn.microsoft.com/en-us/library/cc294424.aspx>)

QUESTION 81

You develop a Windows Communication Foundation (WCF) service that employees use to access bonus information.

You define the following service contract. (Line numbers are included for reference only.)

```
01 [ServiceContract(SessionMode = SessionMode.Required)]
02 public interface IFinancialService
03 {
04     [OperationContract]
05     string Login(int employeeID, string passwordHash);
06
07     [OperationContract]
08     double GetBonus(int month);
09
10     [OperationContract(IsTerminating = true)]
11     void Logout();
12 }
```

Client application can invoke methods without logging in. You need to ensure that the client applications invoke Login before invoking any other method. You also need to ensure that client applications cannot consume the service after invoking Logout.

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

- A. Replace line 04 with the following code.
`[OperationContract(IsInitiating = false)]`
- B. Replace line 04 with the following code.
`[OperationContract(IsInitiating = true, IsTerminating = true)]`
- C. Replace line 07 with the following code.
`[OperationContract(IsInitiating = false)]`
- D. Replace line 10 with the following code.
`[OperationContract(IsInitiating = false, IsTerminating = true)]`

Correct Answer: CD

Section: (none)

Explanation

Explanation/Reference:

OperationContractAttribute.IsInitiating

Gets or sets a value that indicates whether the method implements an operation that can initiate a session on the server (if such a session exists).

OperationContractAttribute.IsInitiating Property

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.operationcontractattribute.isinitiating.aspx>)

QUESTION 82

You develop a Windows Communication Foundation (WCF) service that interacts with Microsoft Message Queuing (MSMQ). The service requires sessions.

You need to create a custom binding that enables messages sent to the queue to be viewed when you are using a listener tool.

Which binding elements should you use?

- A. `textMessageEncoding` and `msmqTransport` in this order.
- B. `textMessageEncoding` and `msmqIntegrationTransport` in this order.
- C. `msmqTransport` and `textMessageEncoding` in this order.
- D. `msmqIntegrationTransport` and `textMessageEncoding` in this order.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

You can add binding elements by adding the desired `BindingElement` objects to its `Elements` collection. The order in which the binding element is added is very important. The order of adding the binding elements is as follows:

1. Transaction Flow (not required)
2. Reliable Messaging (not required)
3. Message Security (not required)
4. Composite Duplex (not required)
5. Message Encoding (required)
6. Transport Security (not required)
7. Transport (required)

Bindings

(<http://msdn.microsoft.com/en-us/library/ff650874.aspx>)

<textMessageEncoding>

Specifies the character encoding and message versioning used for text-based XML messages.

<msmqTransport>

Causes a channel to transfers messages on the MSMQ transport when it is included in a custom binding.

<textMessageEncoding>

(<http://msdn.microsoft.com/en-us/library/ms731787.aspx>)

<msmqTransport>

(<http://msdn.microsoft.com/en-us/library/ms731307.aspx>)

Message Queuing (MSMQ)

([http://msdn.microsoft.com/en-us/library/windows/desktop/ms711472\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms711472(v=vs.85).aspx))

QUESTION 83

You are creating a Windows Communication Foundation (WCF) service. The service endpoints change frequently.

On the service, you add a new ServiceDiscoveryBehavior to the Behaviors collection of the ServiceHost Description property.

You need to ensure that client applications can communicate with the service and discover changes to the service endpoints.

What should you do?

- A. Add a new ServiceDiscoveryBehavior to the Behaviors collection in the client application.
- B. Add a new AnnouncementClient to the Behaviors collection in the client application.
- C. Use the FindCriteria class and the UdpDiscoveryEndpoint class to set up the binding in the client application.
- D. Use the DiscoveryProxy class and the EndpointDiscoveryMetadata class to set up the binding in the client application.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

FindCriteria Class

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.discovery.findcriteria.aspx>)

UdpDiscoveryEndpoint Class

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.discovery.udpdiscoveryendpoint.aspx>)

Configuring Discovery in a Configuration File

(<http://msdn.microsoft.com/en-us/library/dd456792.aspx>)

QUESTION 84

You are developing an application to update a users social status. You need to consume the service using Windows Communication Foundation (WCF).

The client configuration is as follow.



<http://www.gratisexam.com/>

```
<system.serviceModel>
  <bindings>
    <webHttpBinding>
      <binding name="SocialConfig">
        <security mode="TransportCredentialOnly">
```

```

        <transport clientCredentialType="Basic" realm="Social API" />
    </security>
    </binding>
</webHttpBinding>
</bindings>
<client>
    <endpoint address="http://contoso.com" binding="webHttpBinding"
        bindingConfiguration="SocialConfig"
        contract="ISocialStatus" name="SocialClient" />
</client>
</system.serviceModel>

```

The service contract is defined as follows.

```

[ServiceContract]
public interface ISocialStatus
{
    [OperationContract]
    [WebInvoke(UriTemplate = "/statuses/update.xml?status={text}")]
    void UpdateStatus(string text);
}

```

Which code segment should you use to update the social status?

- A. `using (WebChannelFactory<ISocialStatus> factory = new WebChannelFactory<ISocialStatus>("SocialClient"))`
- ```

{
 factory.Credentials.UserName.UserName = user.Name;
 factory.Credentials.UserName.Password = user.Password;
 ISocialStatus socialChannel = factory.CreateChannel();
 socialChannel.UpdateStatus(newStatus);
}

```
- B. `using (ChannelFactory<ISocialStatus> factory = new WebChannelFactory<ISocialStatus>(typeof(ISocialStatus)))`
- ```

{
    factory.Credentials.UserName.UserName = user.Name;
    factory.Credentials.UserName.Password = user.Password;
    ISocialStatus socialChannel = factory.CreateChannel();
    socialChannel.UpdateStatus(newStatus);
}

```
- C. `using (ChannelFactory<ISocialStatus> factory = new ChannelFactory<ISocialStatus>("POST"))`
- ```

{
 factory.Credentials.Windows.ClientCredential.UserName = user.Name;
 factory.Credentials.Windows.ClientCredential.SecurePassword.SetAt(0,
Convert.ToChar(user.Password));
 ISocialStatus socialChannel = factory.CreateChannel();
 socialChannel.UpdateStatus(newStatus);
}

```
- D. `using (WebChannelFactory<ISocialStatus> factory = new WebChannelFactory<ISocialStatus>(typeof(ISocialClient)))`
- ```

{
    factory.Credentials.Windows.ClientCredential.UserName = user.Name;
    factory.Credentials.Windows.ClientCredential.SecurePassword.SetAt(0,
Convert.ToChar(user.Password));
    ISocialStatus socialChannel = factory.CreateChannel();
    socialChannel.UpdateStatus(newStatus);
}

```

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:**WebChannelFactory<TChannel>() Class**

(<http://msdn.microsoft.com/en-us/library/bb908674.aspx>)

ChannelFactory<TChannel> Class

(<http://msdn.microsoft.com/en-us/library/ms576132.aspx>)

WebChannelFactory<TChannel>() Initializes a new instance of the WebChannelFactory<TChannel> class.

WebChannelFactory<TChannel>(Binding) Initializes a new instance of the WebChannelFactory<TChannel> class.

WebChannelFactory<TChannel>(ServiceEndpoint) Initializes a new instance of the WebChannelFactory<TChannel> class.

WebChannelFactory<TChannel>(String) Initializes a new instance of the WebChannelFactory<TChannel> class.

WebChannelFactory<TChannel>(Type) Initializes a new instance of the WebChannelFactory<TChannel> class.

WebChannelFactory<TChannel>(Uri) Initializes a new instance of the WebChannelFactory<TChannel> class with the specified Uri.

WebChannelFactory<TChannel>(Binding, Uri) Initializes a new instance of the WebChannelFactory<TChannel> class with the specified binding and Uri.

WebChannelFactory<TChannel>(String, Uri) Initializes a new instance of the

WebChannelFactory<TChannel> class with the specified endpoint configuration and Uri.

QUESTION 85

You are developing a client application that uses the following code to consume a Windows Communication Foundation (WCF) service.

(Line numbers are included for reference only.)

```
01 BasicHttpBinding myBinding = new BasicHttpBinding();
02 EndpointAddress myEndpointAddress = new EndpointAddress("http://contoso.com/
TaxService.svc");
03 ...
04 ITaxService client = channelFactory.CreateChannel();
05 string data = client.GetData(1);
```

You need to consume the service. Which code segment should you insert at line 03?

- A. `var channelFactory = new ChannelFactory<ITaxService>();`
- B. `var channelFactory = new ChannelFactory<ITaxService>(myBinding);`
- C. `var channelFactory = new ChannelFactory<ITaxService>(myBinding, myEndpointAddress);`
- D. `var channelFactory = new ChannelFactory<ITaxService>("http://contoso.com/TaxService.svc");`

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:**ChannelFactory<TChannel> Class**

(<http://msdn.microsoft.com/en-us/library/ms576132.aspx>)

ChannelFactory<TChannel>() Initializes a new instance of the ChannelFactory<TChannel> class.

ChannelFactory<TChannel>(Binding) Initializes a new instance of the ChannelFactory<TChannel> class.

ChannelFactory<TChannel>(ServiceEndpoint) Initializes a new instance of the

ChannelFactory<TChannel> class that produces channels with a specified endpoint.

ChannelFactory<TChannel>(String) Initializes a new instance of the ChannelFactory<TChannel> class with a specified endpoint configuration name.

ChannelFactory<TChannel>(Type) Initializes a new instance of the ChannelFactory<TChannel> class.

ChannelFactory<TChannel>(Binding, EndpointAddress) Initializes a new instance of the

ChannelFactory<TChannel> class with a specified binding and endpoint address.

ChannelFactory<TChannel>(Binding, String) Initializes a new instance of the ChannelFactory<TChannel>

class with a specified binding and remote address.

ChannelFactory<TChannel>(String, EndpointAddress) Initializes a new instance of the ChannelFactory<TChannel> class associated with a specified name for the endpoint configuration and remote address.

Delegation and Impersonation with WCF

([http://msdn.microsoft.com/en-us/library/ms730088\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/ms730088(v=vs.90).aspx))

Example:

```
public class HelloService : IHelloService
{
    [OperationBehavior(Impersonation = ImpersonationOption.Required)]
    public string Hello(string message)
    {
        WindowsIdentity callerWindowsIdentity =
        ServiceSecurityContext.Current.WindowsIdentity;
        if (callerWindowsIdentity == null)
        {
            throw new InvalidOperationException
            ("The caller cannot be mapped to a Windows identity.");
        }
        using (callerWindowsIdentity.Impersonate())
        {
            EndpointAddress backendServiceAddress = new EndpointAddress
            ("http://localhost:8000/ChannelApp");
            // Any binding that performs Windows authentication of the client
            can be used.
            ChannelFactory<IHelloService> channelFactory = new
            ChannelFactory<IHelloService>(new NetTcpBinding(), backendServiceAddress);
            IHelloService channel = channelFactory.CreateChannel();
            return channel.Hello(message);
        }
    }
}
```

QUESTION 86

You are developing a client application that consumes a Windows Communication Foundation (WCF) service.

You use the svcutil.exe utility to create a proxy for the service. You use the svcutil.exe switches that generate asynchronous calls.

GetFlight is a service operation that takes no parameters and returns a string.

The GetFlightCallback method must be called when the service operation returns.

You create an instance of the client proxy with the following code:

```
var client = new TravelServiceClient();
```

You need to ensure that a callback is received when the GetFlight operation is called asynchronously. Which code segment should you use?

- A. `client.BeginGetFlight(GetFlightCallback, null);`
`client.GetFlight();`
- B. `client.GetFlight();`
`client.BeginGetFlight(GetFlightCallback, null);`
- C. `client.GetFlightCompleted += new EventHandler<GetFlightCompletedEventArgs>`
`(GetFlightCallback);`
`client.GetFlightAsync();`
- D. `IAsyncResult asyncResult = client.BeginGetFlight(GetFlightCallback, client);`
`client.EndGetFlight(asyncResult);`

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

```
// AddAsync
double value1 = 100.00D;
double value2 = 15.99D;
client.AddCompleted += new EventHandler<AddCompletedEventArgs>(AddCallback);
client.AddAsync(value1, value2);
Console.WriteLine("Add({0},{1})", value1, value2);
```

How to: Call WCF Service Operations Asynchronously

(<http://msdn.microsoft.com/en-us/library/ms730059.aspx>)

QUESTION 87

You are developing a Windows Communication Foundation (WCF) service.

The service operation takes a customer number as the only argument and returns information about the customer.

The service requires a security token in the header of the message. You need to create a message contract for the service.

Which code segment should you use?

- A. [ServiceContract]
public interface IService
{
 [OperationContract]
 CustomerInformation GetCustomerInformation(Header header, int customerNumber);
}
- [DataContract]
public class CustomerInformation
{
 ...
}
- [MessageContract]
public class Header
{
 [MessageHeader]
 public string SecurityTag;
}
- B. [ServiceContract]
public interface IService
{
 [OperationContract]
 CustomerInformation GetCustomerInformation(Header header, int customerNumber);
}
- [MessageContract]
public class CustomerInformation
{
 ...
}
- [MessageContract]
public class Header
{
 [MessageHeader]
 public string SecurityTag;
}
- C. [ServiceContract]
public interface IService
{
 [OperationContract]
 CustomerInformation GetCustomerInformation(CustomerNumber request);
}

```

    }

    [DataContract]
    public class CustomerInformation
    {
        ...
    }

    [MessageContract]
    public class CustomerNumber
    {
        [MessageHeader]
        public string SecurityTag;

        [MessageBodyMember]
        public int CustomerNumberElement;
    }
D. [ServiceContract]
public interface IService
{
    [OperationContract]
    CustomerInformation GetCustomerInformation(CustomerNumber request);
}

[MessageContract]
public class CustomerInformation
{
    ...
}

[MessageContract]
public class CustomerNumber
{
    [MessageHeader]
    public string SecurityTag;

    [MessageBodyMember]
    public int CustomerNumberElement;
}

```

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

Using Message Contracts

(<http://msdn.microsoft.com/en-us/library/ms730255.aspx>)

QUESTION 88

Your company has a Windows Communication Foundation (WCF) service at the URL <http://services.contoso.com/OrderLookupService.svc>.

The `<system.serviceModel>` section of the configuration file is as follows. (Line numbers are included for reference only.)

```

01 <system.serviceModel>
02   <behaviors>
03     <serviceBehaviors>
04       <behavior>
05         <serviceDebug includeExceptionDetailInFaults="false" />
06         ...
07       </behavior>
08     </serviceBehaviors>
09   </behaviors>
10   <serviceHostingEnvironment multipleSiteBindingsEnabled="true" />
11 </system.serviceModel>

```

You need to ensure that the service publishes the WSDL description at <http://services.contoso.com/OrderLookupService.svc?wsdl>.
What should you do?

- A. Change the `serviceDebug` element at line 05 as follows.
`<serviceDebug includeExceptionDetailInFaults="true" />`
- B. Insert the following element at line 06.
`<serviceDiscovery>
 <announcementEndpoints>
 <endpoint name="wsdlAnnouncement" kind="udpAnnouncementEndpoint" />
 </announcementEndpoints>
</serviceDiscovery>`
- C. Insert the following element at line 06.
`<serviceMetadata httpGetEnabled="true" />`
- D. Insert the following element at line 06.
`<serviceMetadata httpGetEnabled="false" />`

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

<serviceMetadata>

(<http://msdn.microsoft.com/en-us/library/ms731317.aspx>)

httpGetEnabled A Boolean value that specifies whether to publish service metadata for retrieval using an HTTP/Get request.

The default is false. If the `httpGetUrl` attribute is not specified, the address at which the metadata is published

is the service address plus a "?wsdl". For example, if the service address is "http://localhost:8080/CalculatorService",

the HTTP/Get metadata address is "http://localhost:8080/CalculatorService?wsdl".

If this property is false, or the address of the service is not based on HTTP or HTTPS, "?wsdl" is ignored.

QUESTION 89

A Windows Communication Foundation (WCF) service handles online order processing for your company. You discover that many requests are being made with invalid account numbers.

You create a class named `AccountNumberValidator` that has a method named `Validate`.

Before the message is processed, you need to validate account numbers with `AccountNumberValidator` and reject messages with invalid account numbers.

You create a new class that implements the `IParameterInspector` interface. Which code segment should you use?

- A.

```
public void AfterCall(string operationName, object[] outputs, object
returnValue, object correlationState)
{
    String accountNumber = GetAccountNumber(outputs);
    var validator = new AccountNumberValidator();
    if(validator.Validate(accountNumber))
    {
        throw new FaultException();
    }
}
public object BeforeCall(string operationName, object[] inputs)
{
    return null;
}
```
- B.

```
public void AfterCall(string operationName, object[] outputs, object
```

```

returnValue, object correlationState)
{
    return;
}
public object BeforeCall(string operationName, object[] inputs)
{
    string accountNumber = GetAccountNumber(inputs);
    var validator = new AccountNumberValidator();
    if (!validator.Validate(accountNumber))
    {
        throw new FaultException();
    }
}

```

C.

```

public void AfterCall(string operationName, object[] outputs, object
returnValue, object correlationState)
{
    String accountNumber = GetAccountNumber(outputs);
    var validator = new AccountNumberValidator();
    if( !validator.Validate(accountNumber))
    {
        returnValue = new FaultException();
    }
}
public object BeforeCall(string operationName, object[]inputs)
{
    return null;
}

```

D.

```

public void AfterCall(string operationName, object[] outputs, object
returnValue, object correlationState)
{
    return;
}
public object BeforeCall(string operationName, object[]inputs)
{
    string accountNumber = GetAccountNumber(inputs);
    var validator = new AccountNumberValidator();
    if (!validator.Validate(accountNumber))
    {
        return new FaultException();
    }
}

```

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

Darth was right, there was no correct answer, the IParameterInspector does not expect exceptions as return values, they must be thrown.

I changed option B to represent a correct answer, its previous content was obviously wrong anyway.

AfterCall:

Called after client calls are returned and before service responses are sent.

On outbound calls from a client, the inspector is invoked before the request contents are serialized and sent to the service.

The inspector is also called after the response has been deserialized but before the return values have been dispatched to the proxy method.

On inbound calls to a service, the inspector is invoked after parameters are deserialized but before they are dispatched to the service operation.

BeforeCall:

Called before client calls are sent and after service responses are returned.

On outbound calls from a client, the inspector is invoked before the request contents are serialized and sent to the service.

The inspector is also called after the response has been deserialized but before the return values have been dispatched to the proxy method.

On inbound calls to a service, the inspector is invoked after the request contents are deserialized and dispatched to the service operation and before the response contents are serialized and sent to the client.

QUESTION 90

You develop a Windows Communication Foundation (WCF) service to generate reports.

Client applications call the service to initiate report generation but do not wait for the reports to be generated.

The service does not provide any status to the client applications.

The service class is defined as follows. (Line numbers are included for reference only.)

```
01 [ServiceContract]
02 public class ReportGeneratorService
03 {
04     ...
05     private int GenerateReports(int clientID)
06     {
07         ...
08         return 0;
09     }
10 }
```

You need to ensure that client applications can initiate reports without waiting for status.

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

- A. Insert the following code at line 04.
[OperationContract(IsOneWay=true)]
- B. Insert the following code at line 04.
[OperationContract(AsyncPattern=false)]
- C. At line 05, change the GenerateReports method from private to public.
- D. Remove line 08. At line 05, change the return type of GenerateReports method to void.

Correct Answer: AD

Section: (none)

Explanation

Explanation/Reference:

How to: Create a One-Way Contract

(<http://msdn.microsoft.com/en-us/library/ms733035.aspx>)

One-way methods invoke operations on a Windows Communication Foundation (WCF) service from a client but do not expect a reply.

QUESTION 91

You are creating a client application and configuring it to call a Windows Communication Foundation (WCF) service.

When the application is deployed, it will be configured to send all messages to a WCF routing service.

You need to ensure that the application can consume the target service after the application is deployed.

What should you do?

- A. In the client application, add a service reference to the router service.
In the client binding configuration, specify the address of the router service.
- B. In the client application, add a service reference to the target service.

In the client binding configuration, specify the address of the target service.

- C. In the client application, add a service reference to the router service.
In the client binding configuration, specify the address of the target service.
- D. In the client application, add a service reference to the target service.
In the client binding configuration, specify the address of the router service.

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 92

You create a service and deploy it on a network in a building named Building1. You will deploy the service to Building2.

The service in Building1 is configured using the following discovery scopes.

```
<scopes>
  <add scope="http://contoso.com/Chicago/Building1" />
  <add scope="ldap:///ou=Building1,ou=Chicago,o=contoso,c=us" />
</scopes>
```

The service in Building2 will be configured using the following discovery scopes.

```
<scopes>
  <add scope="http://contoso.com/Chicago/Building2" />
  <add scope="ldap:///ou=Building2,ou=Chicago,o=contoso,c=us" />
</scopes>
```

You need to ensure that the client application can discover the service in Building1 or the service in Building2.

Which scopes should you add to the client configuration file?

- A.

```
<scopes>
  <add scope="http://contoso.com/Chicago/*" />
</scopes>
```
- B.

```
<scopes>
  <add scope="http://contoso.com/Chicago" />
</scopes>
```
- C.

```
<scopes>
  <add scope="ldap:///ou=Building,ou=Chicago,o=contoso,c=us" />
</scopes>
```
- D.

```
<scopes>
  <add scope="ldap:///ou=*,o=contoso,c=us" />
</scopes>
```

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

Configuring Discovery in a Configuration File

(<http://msdn.microsoft.com/en-us/library/dd456792.aspx>)

QUESTION 93

Your company has an existing Windows Communication Foundation (WCF) service that allows business

partners to place orders.

The service uses netMsmqBinding. You find that processing every order in its own transaction is causing a delay.

You need to ensure that the service is configured to process multiple orders in one transaction. What should you do?

- A. Use <serviceThrottling> service behavior and set the maxConcurrentCalls attribute.
- B. Use <transactedBatching> endpoint behavior and set the maxBatchSize attribute.
- C. Use <dispatcherSynchronizationBehavior> endpoint behavior and set the maxPendingReceives attribute.
- D. Use <synchronousReceive> endpoint behavior.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

<transactedBatching>

(<http://msdn.microsoft.com/en-us/library/ms731319.aspx>)

<transactedBatching> A transport that is configured with transaction batching attempts to batch several receive operations into one transaction.

By doing so, the relatively high cost of creating a transaction and committing it in every receive operation is avoided.

<serviceThrottling> Specifies the throttling mechanism of a Windows Communication Foundation (WCF) service.

Throttling controls place limits on the number of concurrent calls, instances, or sessions to prevent over-consumption of resources.

A trace is written every time the value of attributes is reached. The first trace is written as a warning.

QUESTION 94

You need to modify a client application that consumes a Windows Communication Foundation (WCF) service.

The service metadata is no longer available. You need to modify the previously generated proxy to include asynchronous calls to the service.

What should you do?

- A. Update the service reference with the Generate asynchronous operations option.
- B. Create a partial class for the previously generated proxy and include the new asynchronous methods.
- C. Create a class with the same name as the previously generated proxy and add the new asynchronous methods.
Add the new class to a namespace that is different from the original proxy.
- D. Create a class with the same name as the previously generated proxy and add the new asynchronous methods as partial methods.
Add the new class to a namespace that is different from the original proxy.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 95

You are developing a Windows Communication Foundation (WCF) service that returns location information for authorized law enforcement agencies. The service contract is as follows:

```
[OperationContract]
public interface IMappingService
{
    [OperationContract]
    long[] GetLocationCoordinates(String cityName);
    [OperationContract]
    long[] GetLocationOfCitizen(String ssn);
}
```

Users are authenticated and impersonated. The system uses ASP.NET roles. The members of law enforcement are members of the LawEnforcement role. You need to ensure that only members of the LawEnforcement role can call these methods. What are two possible ways to achieve this goal? (Each correct answer presents a complete solution. Choose two.)

- A. Add a PrincipalPermissionAttribute to each method that should be available only to members of law enforcement. Set its SecurityAction to Demand and set the role equal to LawEnforcement.
- B. Use the CurrentPrincipal property of the thread. Call the IsInRole method specifying LawEnforcement as a parameter.
- C. Create a GenericPrincipal specifying Thread.CurrentPrincipal.Identity as the IIdentityParameter and LawEnforcement as the only value for the Roles parameter.
- D. At the beginning of each method, enumerate each ClaimSet in a new WindowsClaimSet. Use the FindClaims method to locate a claim type named Role with a right named LawEnforcement.

Correct Answer: AB

Section: (none)

Explanation

Explanation/Reference:

Thread.CurrentPrincipal Property

Gets or sets the thread's current principal (for role-based security).

Thread.CurrentPrincipal Property

(<http://msdn.microsoft.com/en-us/library/system.threading.thread.currentprincipal.aspx>)

To demand user membership:

- Open the Windows Communication Foundation (WCF) code file that contains the implemented service contract code.
- Apply the PrincipalPermissionAttribute attribute to each method that must be restricted to a specific group.
- Set the Action property to Demand and the Role property to the name of the group.

For example:

```
// Only members of the CalculatorClients group can call this method.
[PrincipalPermission(SecurityAction.Demand, Role = "CalculatorClients")]
public double Add(double a, double b)
{
    return a + b;
}
```

How to: Restrict Access with the PrincipalPermissionAttribute Class

([http://msdn.microsoft.com/en-us/library/ms731200\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/ms731200(v=vs.90).aspx))

QUESTION 96

You are developing a Windows Communication Foundation (WCF) service.

You must record all available information for the first 1,000 messages processed, even if they are malformed.

You need to configure the message logging section of the configuration file. Which configuration segment

should you use?

- A. `<messageLogging logEntireMessage="true" logMalformedMessages="true" logMessagesAtServiceLevel="true" logMessagesAtTransportLevel="true" maxMessagesToLog="1000" />`
- B. `<messageLogging logMessagesAtServiceLevel="true" logMessagesAtTransportLevel="true" maxMessagesToLog="1000" />`
- C. `<messageLogging logEntireMessage="false" logMessagesAtServiceLevel="true" logMessagesAtTransportLevel="false" maxMessagesToLog="1000" />`
- D. `<messageLogging logMalformedMessages="true" logMessagesAtServiceLevel="true" logMessagesAtTransportLevel="false" maxMessagesToLog="1000" />`

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

Configuring Message Logging

(<http://msdn.microsoft.com/en-us/library/ms730064.aspx>)

MessageLoggingElement properties:

LogEntireMessage Gets or sets a value that specifies whether the entire message (message header and body) is logged.

logMalformedMessages

LogMessagesAtServiceLevel Gets or sets a value that specifies whether messages are traced at the service level (before encryption and transport-related transforms).

LogMessagesAtTransportLevel Gets or sets a value that specifies whether messages are traced at the transport level.

QUESTION 97

You are developing a Windows Communication Foundation (WCF) service.

The service configuration file has a <System.Diagnostics> element defined.

You need to ensure that all security audit information, trace logging, and message logging failures are recorded.

Which configuration segment should you add to the <System.Diagnostics> element?

- A. `<sources>
 <source name="System.ServiceModel" switchValue="Information, ActivityTracing" propagateActivity="true">
 <listeners>
 <add name="xml" />
 </listeners>
 </source>
 <source name="System.ServiceModel.MessageLogging" propagateActivity="true">
 <listeners>
 <add name="text" />
 </listeners>
 </source>
</sources>`
- B. `<sources>
 <source name="System.ServiceModel" switchValue="Information,`

```

ActivityTracing" propagateActivity="true" />
  <source name="System.ServiceModel.MessageLogging"
propagateActivity="true" />
</sources>

```

C. <sources>

```

  <source name="System.ServiceModel" switchValue="Information,
ActivityTracing" propagateActivity="true">
    <listeners>
      <add name="xml" />
    </listeners>
  </source>
  <source name="System.ServiceModel.MessageLogging"
propagateActivity="true">
    <listeners>
      <add name="xml" />
    </listeners>
  </source>
</sources>
<sharedListeners>
  <add name="xml" type="System.Diagnostics.XmlWriterTraceListener"
initializeData="& " />
</sharedListeners>

```

D. <sources>

```

  <source name="System.ServiceModel" switchValue="Information,
ActivityTracing" propagateActivity="true" />
  <source name="System.ServiceModel.MessageLogging"
propagateActivity="true" />
</sources>

<sharedListeners>
  <add name="xml" type="System.Diagnostics.XmlWriterTraceListener"
initializeData="& " />
</sharedListeners>

```

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 98

You are developing a Windows Communication Foundation (WCF) service that contains the following service contract.

```

[ServiceContract()]
public interface IPaymentService
{
    [OperationContract()]
    void RecordPayments(Person person);
}

public class Person
{ ... }
public class Employee : Person
{ ... }
public class Customer : Person
{ ... }

```

You need to ensure that RecordPayments can correctly deserialize into an Employee or a Customer object. What should you do?

A. Add the following KnownType attribute to the Employee class and to the Customer class.

```
[KnownType(GetType(Person))]
```

- B. Implement the `IExtensibleDataObject` interface in the `Person` class.
- C. Implement the `IExtension(ofType(T))` interface in the `Person` class.
- D. Add the following `KnownType` attributes to the `Person` class.
[KnownType(GetType(Employee))]
[KnownType(GetType(Customer))]

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

Data Contract Known Types

(<http://msdn.microsoft.com/en-us/library/ms730167.aspx>)

The `KnownTypeAttribute` class allows you to specify, in advance, the types that should be included for consideration during deserialization.

QUESTION 99

You are developing a Windows Communication Foundation (WCF) service to provide an in-memory cache. The following code is part of your solution. (Line numbers are included for reference only.)

```
01
02 public interface IInMemoryCacheService
03 {
04     [OperationContract()]
05     string GetCachedItem(string key);
06
07     [OperationContract()]
08     void CacheItem(string key, string item);
09
10 }
11
12 [ServiceBehavior(InstanceContextMode=InstanceContextMode.Single)]
13 public class CacheService : IInMemoryCacheService
14 {
15
16     Hashtable cache = new Hashtable();
17
18     public string GetCachedItem(string key)
19     {
20         return cache[key].ToString();
21     }
22
23     public void CacheItem(string key, string item)
24     {
25         if (cache.Contains(key))
26             cache.Remove(key);
27         cache.Add(key, item);
28     }
29 }
```

Users report that the cache is getting updated with cache changes of other users.

You need to ensure that each user's cache is maintained and isolated from other users.

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

- A. Insert the following code at line 01.
[ServiceContract(SessionMode=SessionMode.NotAllowed)]
- B. At line 12, replace `InstanceContextMode.Single` with `InstanceContextMode.PerSession`.

C. At line 12, replace `InstanceContextMode.Single` with `InstanceContextMode.PerCall`.

D. Insert the following code at line 01.

```
[ServiceContract(SessionMode=SessionMode.Required)]
```

Correct Answer: BD

Section: (none)

Explanation

Explanation/Reference:

InstanceContextMode enumeration

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.instancecontextmode.aspx>)

PerSession A new InstanceContext object is created for each session.

PerCall A new InstanceContext object is created prior to and recycled subsequent to each call. If the channel does not create a session this value behaves as if it were PerCall.

Single Only one InstanceContext object is used for all incoming calls and is not recycled subsequent to the calls.

If a service object does not exist, one is created.

QUESTION 100

A Windows Communication Foundation (WCF) service is hosted in Microsoft Internet Information Services (IIS).

You are preparing the configuration file for production deployment. You need to set up tracing so that an administrator can adjust

the System.ServiceModel trace level without causing the ASP.NET host application domain to restart.

You need to ensure that tracing is turned off by default. What should you do?

A. Add the following element to the system.serviceModel configuration section:

```
<diagnostics wmiProviderEnabled="true" />
```

Add the following element to the system.diagnostics configuration section:

```
<sources>
  <source name="System.ServiceModel" switchValue="Off">
    <listeners>
      <add initializeData="app_tracelog.svclog"
        type="System.Diagnostics.XmlWriterTraceListener"
        name="ServiceModelTraceListener"
        traceOutputOptions="Timestamp" />
    </listeners>
  </source>
</sources>
```

B. Add the following element to the system.serviceModel configuration section:

```
<diagnostics etwProviderId="{7799e76e-0f8b-407a-a616-3f91ba6072b9}">
```

Add the following element to the system.diagnostics configuration section:

```
<sources>
  <source name="System.ServiceModel" switchValue="Off">
    <listeners>
      <add initializeData="app_tracelog.svclog"
        type="System.Diagnostics.XmlWriterTraceListener"
        name="ServiceModelTraceListener"
        traceOutputOptions="Timestamp" />
    </listeners>
  </source>
</sources>
```

C. Add the following element to the system.serviceModel configuration section:

```
<serviceBehaviors>
  <behavior>
    <serviceDebug />
  </behavior>
</serviceBehaviors>
```

Add the following element to the system.diagnostics configuration section:

```
<sources>
  <source name="System.ServiceModel" switchValue="ActivityTracing">
    <listeners>
      <add initializeData="app_tracelog.svclog"
        type="System.Diagnostics.XmlWriterTraceListener"
        name="ServiceModelTraceListener"
        traceOutputOptions="Timestamp" />
    </listeners>
  </source>
</sources>
```

D. Add the following element to the system.serviceModel configuration section:

```
<serviceBehaviors>
  <behavior>
    <serviceDebug/>
  </behavior>
</serviceBehaviors>
```

Add the following element to the system.diagnostics configuration section:

```
<sources>
  <source name="System.ServiceModel" switchValue="Off">
    <listeners>
      <add initializeData="app_tracelog.svclog"
        type="System.Diagnostics.XmlWriterTraceListener"
        name="ServiceModelTraceListener"
        traceOutputOptions="Timestamp" />
    </listeners>
  </source>
</sources>
```

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

Enabling WMI

A WMI provider is a component that exposes instrumentation at runtime through a WBEM-compatible interface.

It consists of a set of WMI objects that have attribute/value pairs. Pairs can be of a number of simple types. Management tools can connect to the services through the interface at runtime.

WCF exposes attributes of services such as addresses, bindings, behaviors, and listeners.

The built-in WMI provider can be activated in the configuration file of the application.

This is done through the wmiProviderEnabled attribute of the <diagnostics> Element in the system.ServiceModel element section, as shown in the following sample configuration.

```
<system.serviceModel>
  ...
  <diagnostics wmiProviderEnabled="true" />
  ...
</system.serviceModel>
```

This configuration entry exposes a WMI interface. Management applications can now connect through this interface and access the management instrumentation of the application.

Using Windows Management Instrumentation for Diagnostics

([http://msdn.microsoft.com/en-us/library/ms735120\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/ms735120(v=vs.90).aspx))

QUESTION 101

A client application calls a Windows Communication Foundation (WCF) service with a proxy class that was generated by Visual Studio.

The service binding was changed from wsHttpBinding to basicHttpBinding. The method signatures in the service contract are changed.

After these changes, the client application receives errors when calling methods on the service.

You need to ensure that the client can invoke methods on the service successfully. What should you do?

- A. Update the configuration file to use basicHttpBinding.
- B. Enable anonymous authentication for the service.
- C. Copy the configuration file elements under the <system.serviceModel> element from the service back to the client.
- D. Update the service reference.

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 102

You are configuring services to be discoverable. The services must be discoverable without relying on a central server.

Client applications that consume the services are on a network segment that is separate from the network segment that the services are located on.

A firewall blocks all TCP ports between the two network segments, but allows other protocols to pass through.

You need to ensure that the client applications can discover the services. What should you do?

- A. Use ad-hoc discovery mode over HTTP.
- B. Use ad-hoc discovery mode over UDP.
- C. Use managed discovery mode over HTTP.
- D. Use managed discovery mode over UDP.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

Ad-Hoc and Managed Modes

The Discovery API supports two different modes: Managed and Ad-Hoc. In Managed mode there is a centralized server

called a discovery proxy that maintains information about available services. The discovery proxy can be populated with

information about services in a variety of ways. For example, services can send announcement messages during start up

to the discovery proxy or the proxy may read data from a database or a configuration file to determine what services are available.

How the discovery proxy is populated is completely up to the developer. Clients use the discovery proxy to retrieve information about available services.

When a client searches for a service it sends a Probe message to the discovery proxy and the proxy determines whether any

of the services it knows about match the service the client is searching for. If there are matches the discovery proxy sends a ProbeMatch

response back to the client. The client can then contact the service directly using the service information returned from the proxy.

The key principle behind Managed mode is that the discovery requests are sent in a unicast manner to one authority, the discovery proxy.

The .NET Framework contains key components that allow you to build your own proxy. Clients and services can locate the proxy by multiple methods:

- The proxy can respond to ad-hoc messages.

- The proxy can send an announcement message during start up.
- Clients and services can be written to look for a specific well-known endpoint.

In Ad-Hoc mode, there is no centralized server. All discovery messages such as service announcements and client requests are sent in a multicast fashion. By default the .NET Framework contains support for Ad-Hoc discovery over the UDP protocol.

For example, if a service is configured to send out a Hello announcement on start up, it sends it out over a well-known, multicast address using the UDP protocol. Clients have to actively listen for these announcements and process them accordingly.

When a client sends a Probe message for a service it is sent over the network using a multicast protocol. Each service that receives the request determines whether it matches the criteria in the Probe message and responds directly to the client with a ProbeMatch message if the service matches the criteria specified in the Probe message.

WCF Discovery Overview

(<http://msdn.microsoft.com/en-us/library/dd456791.aspx>)

QUESTION 103

You are developing a Windows Communication Foundation (WCF) client application. The client application contains the following code.

```
[ServiceContract]
public interface ISocialStatus
{
    [OperationContract]
    [WebInvoke(UriTemplate = "/statuses/update.xml?status-{text}")]
    void UpdateStatus(string text);
}
public class SocialClient : ClientBase<ISocialStatus>, ISocialStatus
{
    ...
}
```

The configuration file contains the following lines.

```
<system.serviceModel>
  <client>
    <endpoint name="SocialClient" address="http://contoso.com"
      binding="webHttpBinding" contract="SocialApp.ISocialStatus"
      bindingConfiguration="BindingConfig" />
  </client>
  <bindings />
</system.serviceModel>
```

You need to ensure that the service is consumed. Which code segment should you use?

- var client = new SocialClient("SocialClient");
client.Endpoint.Behaviors.Add(new WebHttpBehavior());
- var client = new SocialClient("SocialClient");
client.Endpoint.Behaviors.Add(new WebScriptEnablingBehavior());
- var client = new SocialClient("POST");
client.Endpoint.Behaviors.Add(new WebHttpBehavior());
- var client = new SocialClient("POST");
client.Endpoint.Behaviors.Add(new WebScriptEnablingBehavior());

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

ClientBase<TChannel> Class

(<http://msdn.microsoft.com/en-us/library/ms576141.aspx>)

ClientBase<TChannel> Class

Provides the base implementation used to create client objects that can call services.

Example:

```
public partial class SampleServiceClient : System.ServiceModel.ClientBase<ISampleService>,
ISampleService
{
    public SampleServiceClient()
    {
    }

    public SampleServiceClient(string endpointConfigurationName) :
        base(endpointConfigurationName)
    {
    }

    public SampleServiceClient(string endpointConfigurationName, string remoteAddress) :
        base(endpointConfigurationName, remoteAddress)
    {
    }

    public SampleServiceClient(string endpointConfigurationName, System.ServiceModel.EndpointAddress
remoteAddress) :
        base(endpointConfigurationName, remoteAddress)
    {
    }

    public SampleServiceClient(System.ServiceModel.Channels.Binding binding,
System.ServiceModel.EndpointAddress remoteAddress) :
        base(binding, remoteAddress)
    {
    }

    public string SampleMethod(string msg)
    {
        return base.Channel.SampleMethod(msg);
    }
}
```

QUESTION 104

You are developing a new version of an existing message contract named CustomerDetailsVersion1. The new version of the message contract must add a Department field of type String to the SOAP header. You create a new class named CustomerDetailsVersion2 that inherits from CustomerDetailsVersion1.

You need to ensure that all client applications can consume the service. Which code segment should you use?

- A. [MessageContract]
- ```
public class CustomerDetailsVersion2 : CustomerDetailsVersion1
{
 [MessageHeader(MustUnderstand = false)]
 public string Department;
}
```
- B. public class CustomerDetailsVersion2 : CustomerDetailsVersion1
- ```
{
```

```
[MessageHeader(MustUnderstand = false)]
public string Department;
}
```

C. [MessageContract]
public class CustomerDetailsVersion2 : CustomerDetailsVersion1
{
 [MessageHeader(MustUnderstand = true)]
 public string Department;
}

D. public class CustomerDetailsVersion2 : CustomerDetailsVersion1
{
 [MessageHeader(MustUnderstand = true)]
 public string Department;
}

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

SoapHeader.MustUnderstand Property

Gets or sets a value indicating whether the SoapHeader must be understood.

true if the XML Web service must properly interpret and process the SoapHeader; otherwise, false. The default is false.

When an XML Web service client adds a SOAP header to an XML Web service method call with the MustUnderstand property set to true, the XML Web service method must set the DidUnderstand property to true; otherwise, a SoapHeaderException is thrown back to the XML Web service client by ASP.NET.

Although both the EncodedMustUnderstand and MustUnderstand properties can be used to set the value of the mustUnderstand attribute within the SOAP header, the MustUnderstand property allows you to set the attribute using a Boolean value.

Data Contract Versioning

(<http://msdn.microsoft.com/en-us/library/ms731138%28v=VS.100%29.aspx>)

Best Practices: Data Contract Versioning

(<http://msdn.microsoft.com/en-us/library/ms733832.aspx>)

QUESTION 105

You are developing a Windows Communication Foundation (WCF) service.

You establish that the largest size of valid messages is 8,000 bytes. You notice that many malformed messages are being transmitted.

Detailed information about whether each message is malformed must be logged.

You need to ensure that this information is saved in XML format so that it can be easily analyzed. What should you add to the service configuration file?

A. <messageLogging
 logMessagesAtServiceLevel="true"
 logMessagesAtTransportLevel="true"
 maxMessagesToLog="1000"
 maxSizeOfMessageToLog="8000" />

B. <messageLogging
 logEntireMessage="true"
 logMalformedMessages="false"
 logMessagesAtServiceLevel="true"

```
logMessagesAtTransportLevel="true"
maxMessagesToLog="1000" />
```

C. <message Logging
logEntireMessage="true"
logMalformedMessages="false"
logMessagesAtServiceLevel="true"
logMessagesAtTransportLevel="true"
maxMessagesToLog="1000"
maxSizeOfMessageToLog="8000" />

D. <messageLogging
logEntireMessage="true"
logMalformedMessages="true"
logMessagesAtServiceLevel="true"
logMessagesAtTransportLevel="true"
maxMessagesToLog="1000"
maxSizeOfMessageToLog="100000" />

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

Configuring Message Logging

(<http://msdn.microsoft.com/en-us/library/ms730064.aspx>)

logMalformedMessages="true" is a key attribute

QUESTION 106

You are developing a Windows Communication Foundation (WCF) service that allows customers to update financial data.

The client applications call the service in a transaction. The service contract is defined as follows. (Line numbers are included for reference only.)

```
01 [ServiceContract]
02 public interface IDataUpdate
03 {
04     [OperationContract]
05     [TransactionFlow(TransactionFlowOption.Mandatory)]
06     void Update(string accountNumber, double amount);
07 }
08
09 ...
10 class UpdateService : IDataUpdate
11 {
12     [OperationBehavior(TransactionScopeRequired = true,
TransactionAutoComplete = true)]
13     public void Update(string accountNumber, double amount)
14     {
15         try
16         {
17             ...
18         }
19         catch(Exception ex)
20         {
21             WriteErrorLog(ex);
22             ...
23         }
24     }
25 }
26
```

Customers report that the transaction completes successfully even if the Update method throws an exception.

You need to ensure that the transaction is aborted if the Update method is not successful. What should you do?

A. Insert the following line at line 22:

```
throw;
```

B. Insert the following line at line 09:

```
[ServiceBehavior(TransactionAutoCompleteOnSessionClose=false)]
```

C. Replace line 12 with the following line:

```
[OperationBehavior(TransactionScopeRequired=true, TransactionAutoComplete=false)]
```

D. Insert the following line at line 09.:

```
[ServiceBehavior(TransactionAutoCompleteOnSessionClose=true)]
```

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

It is needed to throw Exception. Otherwise, it is handled successfully in Catch block.

QUESTION 107

You create a Windows Communication Foundation (WCF) service. It is deployed on Microsoft Internet Information Services (IIS) with an application pool running as Network Service. You enable WMI tracing before launching the service. Your IT support staff adds WMI data collection through ASP.NET WMI tracing.

You need to restrict the collection of WMI data to a privileged account. What should you do in WMI Control in the Computer Management console?

- A. - Select the Root\ServiceModel namespace.
 - Remove Enable account permission for the Network Service account.
 - Add a custom user and grant that user Enable account permission.
- B. - Select the Root\aspnet namespace.
 - Remove Enable account permission for the Network Service account.
 - Add a custom user and grant that user Enable account permission.
- C. - Select the Root\aspnet namespace.
 - Remove Enable account permission for the Local System account.
 - Add a custom user and grant that user Enable account permission.
- D. - Select the Root\Security namespace.
 - Remove Enable account permission for the Local System account.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

Tracing WMI Activity

([http://msdn.microsoft.com/en-us/library/windows/desktop/aa826686\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa826686(v=vs.85).aspx))

Using Windows Management Instrumentation for Diagnostics

(<http://msdn.microsoft.com/en-us/library/ms735120.aspx>)

Security

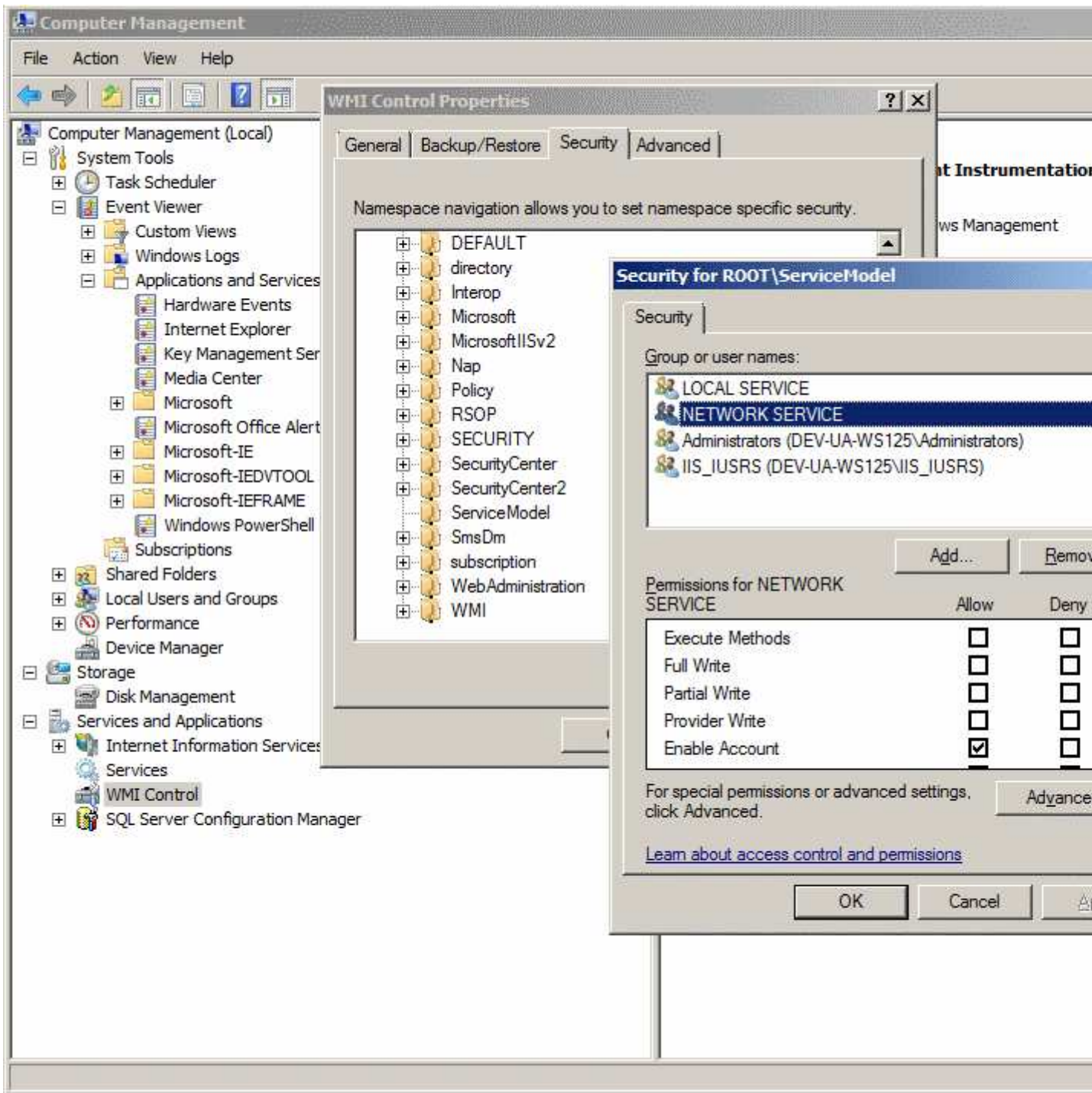
By default, the WCF WMI provider grants "execute method", "provider write", and "enable account"

permission for Administrator, and "enable account" permission for ASP.NET, Local Service and Network Service. In particular, on non-Windows Vista platforms, the ASP.NET account has read access to the WMI ServiceModel namespace. If you do not want to grant these privileges to a particular user group, you should either deactivate the WMI provider (it is disabled by default), or disable access for the specific user group.

In addition, when you attempt to enable WMI through configuration, WMI may not be enabled due to insufficient user privilege. However, no event is written to the event log to record this failure.

To modify user privilege levels, use the following steps:

1. Click Start and then Run and type compmgmt.msc.
2. Right-click Services and Application/WMI Controls to select Properties.
3. Select the Security Tab, and navigate to the Root/ServiceModel namespace. Click the Security button.
4. Select the specific group or user that you want to control access and use the Allow or Deny checkbox to configure permissions.



QUESTION 108

You are creating a Windows Communication Foundation (WCF) service that uses claims-based authorization.

The following code retrieves the correct claim set.

```
var claims = ServiceSecurityContext.Current.AuthorizationContext.ClaimSets[0];
```

You need to validate that the requesting client application has included a valid DNS value in the claim. Which code segment should you use to retrieve the claim for validation?

- A. `claims.FindClaims(ClaimTypes.Dns, Rights.PossessProperty).FirstOrDefault();`
- B. `claims.FindClaims(ClaimTypes.Dns, Rights.Identity).FirstOrDefault();`
- C. `claims.ContainsClaim(Claim.CreateDnsClaim(ClaimTypes.Dns));`
- D. `claims.Equals(ClaimTypes.Dns);`

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

ClaimSet.FindClaims Method

When overridden in a derived class, searches for a Claim object that matches the specified claim type and rights in the ClaimSet.

Identity Gets a string that specifies that the right represents an identity.

PossessProperty Gets a string that specifies that the right represents a property that the entity associated with a claim possesses.

ClaimTypes class

(<http://msdn.microsoft.com/en-us/library/system.identitymodel.claims.claimtypes.aspx>)

Authorization In WCF-Based Services

(<http://msdn.microsoft.com/en-us/magazine/cc948343.aspx>)

Rights Class

(<http://msdn.microsoft.com/en-us/library/system.identitymodel.claims.rights.aspx>)

QUESTION 109

You are creating an application that consumes a Windows Communication Foundation (WCF) service. The service implements the IService contract. The client application contains the CallbackHandler class, which implements IServiceCallback.

You need to ensure that a client proxy is created that can communicate with the service over a duplex channel.

Which code segment should you use?

- A.

```
var handler = new CallbackHandler();
var clientFactory = new DuplexChannelFactory<IService>(new
WSHttpContextBinding());
var client = clientFactory.CreateChannel(new InstanceContext(handler), new
EndpointAddress("..."));
```
- B.

```
var handler = new CallbackHandler();
var clientFactory = new DuplexChannelFactory<IService>(typeof
(CallbackHandler), new WSDualHttpBinding());
var client = clientFactory.CreateChannel(new InstanceContext(handler), new
EndpointAddress("..."));
```
- C.

```
var handler = new CallbackHandler();
var clientFactory = new DuplexChannelFactory<IService>(new WSHttpBinding
());
var client = clientFactory.CreateChannel(new InstanceContext(handler), new
EndpointAddress ("..."));
```
- D.

```
var handler = new CallbackHandler();
var clientFactory = new DuplexChannelFactory<IService>(typeof
(CallbackHandler), new WSDualHttpBinding());
var client = clientFactory.CreateChannel(new EndpointAddress("..."));
```

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

DuplexChannelFactory<TChannel> Class

([http://msdn.microsoft.com/en-us/library/ms576164\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/ms576164(v=vs.90).aspx))

QUESTION 110

You have a Windows Communication Foundation (WCF) service that accepts the following message contract.

```
[MessageContract(WrapperNamespace="http://www.movies.com",
ProtectionLevel=ProtectionLevel.None)]
public class Ticket
{
    [MessageBodyMember(Namespace="http://www.movietheater.com", Order=1)]
    public DateTime ShowTime = DateTime.Now;

    [MessageBodyMember(Namespace="http://www.movietheater.com")]
    public string ReservationName = "Smith";

    [MessageBodyMember(Namespace="http://www.movietheater.com")]
    public int NumberOfSeats = 0;
}
```

You need to ensure that the client sends a SOAP body that is accepted by the service.

- A. `<Ticket xmlns="http://www.movies.com">
 <NumberOfSeats xmlns="http://www.movietheater.com">0</NumberOfSeats>
 <ReservationName xmlns="http://www.movietheater.com" />
 <ShowTime xmlns="http://www.movietheater.com">2010-07-05T00:SI:10.0999304-05:00</ShowTime>
</Ticket>`
- B. `<Ticket xmlns="http://www.movietheater.com">
 <ShowTime xmlns="http://www.movietheater.com">2010-07-05T00:51:10.0999304-05:00</ShowTime>
 <ReservationName xmlns="http://www.movietheater.com" />
 <NumberOfSeats xmlns="http://www.movietheater.com">0</NumberOfSeats>
</Ticket>`
- C. `<Ticket xmlns="http://www.movies.com">
 <ShowTime xmlns="http://www.movietheater.com">2010-07-05T00:51:10.0999304-05:00</ShowTime>
 <NumberOfSeats xmlns="http://www.movietheater.com">0</NumbecOfSeats>
 <ReservationName xmlns="http://www.movietheater.com" />
</Ticket>`
- D. `<Ticket xmlns="http://www.movietheater.com">
 <ShowTime xmlns="http://www.movietheater.com">2010-07-05T00:51:10.0999304-05:00</ShowTime>
 <NumberOfSeats xmlns="http://www.movietheater.com">0</NumberOfSeats>
 <ReservationName xmlns="http://www.movietheater.com" />
</Ticket>`

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

Using Message Contracts

(<http://msdn.microsoft.com/en-us/library/ms730255.aspx>)

MessageBodyMemberAttribute.Order Property

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.messagebodymemberattribute.order.aspx>)

Order of SOAP Body Parts

In some circumstances, you may need to control the order of the body parts. The order of the body elements is alphabetical by default, but can be controlled by the `System.ServiceModel.MessageBodyMemberAttribute.Order` property. This property has the same semantics as the `System.Runtime.Serialization.DataMemberAttribute.Order` property, except for the behavior in

inheritance scenarios

(in message contracts, base type body members are not sorted before the derived type body members).

EXAMPLE:

In the following example, amount would normally come first because it is first alphabetically. However, the Order property puts it into the third position.

```
[MessageContract]
public class BankingTransaction
{
    [MessageHeader] public Operation operation;
    [MessageBodyMember(Order=1)] public Account sourceAccount;
    [MessageBodyMember(Order=2)] public Account targetAccount;
    [MessageBodyMember(Order=3)] public int amount;
}
-----
```

Data Member Order

(<http://msdn.microsoft.com/en-us/library/ms729813.aspx>)

Example:

```
[DataContract]
public class BaseType
{
    [DataMember]
    public string zebra;
}
[DataContract]
public class DerivedType : BaseType
{
    [DataMember(Order = 0)]
    public string bird;
    [DataMember(Order = 1)]
    public string parrot;
    [DataMember]
    public string dog;
    [DataMember(Order = 3)]
    public string antelope;
    [DataMember]
    public string cat;
    [DataMember(Order = 1)]
    public string albatross;
}
```

Output

```
<DerivedType>
  <!-- Zebra is a base data member, and appears first. -->
  <zebra/>

  <!-- Cat has no Order, appears alphabetically first. -->
  <cat/>

  <!-- Dog has no Order, appears alphabetically last. -->
  <dog/>

  <!-- Bird is the member with the smallest Order value -->
  <bird/>

  <!-- Albatross has the next Order value, alphabetically first. -->
  <albatross/>

  <!-- Parrot, with the next Order value, alphabetically last. -->
  <parrot/>
```

```
<!-- Antelope is the member with the highest Order value. Note that Order=2 is skipped -->
<antelope/>
</DerivedType>
```

QUESTION 111

You are debugging a Windows Communication Foundation (WCF) service. The service uses signed and encrypted messages.

You need to configure logging so that you can read the contents of the messages. What should you do?

- A. Set `maxSizeMessagesToLog` to 10
- B. Set `logMessageAtServiceLevel` to `true`.
- C. Set `maxMessagesToLog` to 10.
- D. Set `logMessageAtTransportLevel` to `true`.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

Configuring Message Logging

(<http://msdn.microsoft.com/en-us/library/ms730064.aspx>)

Service Level

Messages logged at this layer are about to enter (on receiving) or leave (on sending) user code.

If filters have been defined, only messages that match the filters are logged.

Otherwise, all messages at the service level are logged. Infrastructure messages (transactions, peer channel, and security)

are also logged at this level, except for Reliable Messaging messages. On streamed messages, only the headers are logged.

In addition, secure messages are logged decrypted at this level.

Transport Level

Messages logged at this layer are ready to be encoded or decoded for or after transportation on the wire.

If filters have been defined, only messages that match the filters are logged. Otherwise, all messages at the transport layer are logged.

All infrastructure messages are logged at this layer, including reliable messaging messages. On streamed messages,

only the headers are logged. In addition, secure messages are logged as encrypted at this level, except if a secure transport such as HTTPS is used.

QUESTION 112

You are developing a client application that consumes a Windows Communication Foundation (WCF) service.

The operation contract is as follows.

```
[OperationContract]
[FaultContract(typeof(SalesFault))]
string GetSales(string saleId);
```

The service configuration file contains the following line in the `serviceBehaviors` section.

```
<behavior>
  <serviceDebug includeExceptionDetailInFaults="True"/>
</behavior>
```

A divide-by-zero exception is not being handled by the service.

You need to ensure that the exception is caught in the client application.

Which type of exception should the client catch?

- A. `TimeoutException`
- B. `FaultException`
- C. `DivideByZeroException`

D. FaultException<SalesFault>

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

Set IncludeExceptionDetailInFaults to true to enable exception information to flow to clients for debugging purposes.

This property requires a binding that supports either request-response or duplex messaging.

In all managed applications, processing errors are represented by Exception objects.

In SOAP-based applications such as applications, methods that implement service operations communicate

error information using SOAP fault messages. Because applications execute under both types of error systems,

any managed exception information that needs to be sent to the client must be converted from exceptions into SOAP faults.

For more information, see Specifying and Handling Faults in Contracts and Services.

During development, you may want your service to also send other exceptions back to the client to assist you in debugging.

This is a development-only feature and should not be employed in deployed services.

To facilitate debugging development, set the IncludeExceptionDetailInFaults to true either in code or using an application configuration file.

When enabled, the service automatically returns safer exception information to the caller.

These faults appear to the client as FaultException<TDetail> objects of type ExceptionDetail.

Important

Setting IncludeExceptionDetailInFaultsto true enables clients to obtain information about internal service method exceptions;

it is only recommended as a way of temporarily debugging a service application. In addition, the WSDL for a method that returns

unhandled managed exceptions in this way does not contain the contract for the FaultException<TDetail> of type ExceptionDetail.

Clients must expect the possibility of an unknown SOAP fault to obtain the debugging information properly.

QUESTION 113

You are developing an application that performs file backups in the background. The background application will host a Windows Communication Foundation (WCF) service and must be active as soon as the machine is running.

The background application will not have a user interface. A front-end application will consume the WCF service and will provide the user interface.

You need to choose a host environment for the WCF service. Which hosting environment should you choose?

A. Microsoft Internet Information Services (IIS) 6.0

B. Windows Process Activation Services (WAS)

C. A Windows Forms application

D. A Windows Service

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

Hosting Services

(<http://msdn.microsoft.com/en-us/library/ms730158.aspx>)

Hosting Options:

1. Self-Hosting in a Managed Application (self-hosted)
2. Managed Windows Services
3. Internet Information Services (IIS)
4. Windows Process Activation Service (WAS)

QUESTION 114

You are developing a Windows Communication Foundation (WCF) service. Client applications require reliable sessions to access the service.

Users report that they receive `ServerTooBusyException` errors when their client application tries to establish a session.

You need to ensure that the service can handle a minimum of 30 client connection requests. Which `ReliableSessionBindingElement` property should you configure?

- A. `MaxRetryCount`
- B. `MaxTransferWindowSize`
- C. `MaxPendingChannels`
- D. `InactivityTimeout`

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

ReliableSessionBindingElement Class

Represents the binding element that can produce the sending and receiving channels required for a reliable session between endpoints.

MaxPendingChannels property

Channels are pending when they are waiting to be accepted. Once that limit is reached, no channels are created and put in pending mode until this number goes down (by accepting pending channels).

This is a limit for each listener.

When the threshold is reached and a remote application tries to establish a new reliable session, the request is denied and the open operation that prompted this faults.

MaxPendingChannels Gets or sets the largest number of channels that can be pending during the reliable session.

MaxRetryCount Gets or sets the maximum number of times that a message attempts to be transferred during the reliable session.

MaxTransferWindowSize Gets or sets the largest number of messages that can exist in either the send buffer or the receive buffer.

InactivityTimeout Gets or sets an interval of time that a service remains inactive before closing.

ReliableSessionBindingElement Class

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.channels.reliableessionbindingelement.aspx>)

QUESTION 115

You are developing a Windows Communication Foundation (WCF) service.

You enable message logging, trace listeners, activity propagation, and tracing on the trace sources.

You have the following code segment in the client application. (Line numbers are included for reference only.)

```
01     Guid multiCallActivityId = Guid.NewGuid();
02     TraceSource ts = new TraceSource("Multicall");
03     Trace.CorrelationManager.ActivityId = multiCallActivityId;
04
```

You encounter errors when your client application consumes the service.

You need to ensure that your client application can correlate tracing information with the service.

Which code segment should you add at line 04?

- A.

```
ts.TraceEvent(TraceEventTypes.Start, 0, "Calling first service");
ts.TraceTransfer(100, "Transferring...", multiCallActivityId);
...
ts.TraceEvent(TraceEventTypes.Stop, 0, "Return from first service.");
```
- B.

```
Trace.CorrelationManager.StartLogicalOperation("1");
...
Trace.CorrelationManager.StopLogicalOperation();
```
- C.

```
ts.TraceEvent(TraceEventTypes.Start, 0, "Calling first service");
ts.TraceTransfer(100, "Transferring...", Guid.NewGuid());
...
ts.TraceEvent(TraceEventTypes.Stop, 0, "Return from first service.");
```
- D.

```
ts.TraceEvent(TraceEventTypes.Start, 0, "Calling first service");
Trace.CorrelationManager.StartLogicalOperation("1");
...
ts.TraceEvent(TraceEventTypes.Stop, 0, "Return from first service.");
```

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

Propagating the Activity ID to A Service

When the activity ID is propagated across endpoints, the message receiver emits a Start and Stop traces with that (propagated) activity ID. Therefore, there is a Start and Stop trace with that gAId from each trace source.

If the endpoints are in the same process and use the same trace source name, multiple Start and Stop with the same lAId (same gAId, same trace source, same process) are created.

If you set the propagateActivity attribute to true for the System.ServiceModel trace source in both the client and service configuration files, the service processing for the Add request occurs in the same activity as the one defined in the client.

If the service defines its own activities and transfers, the service traces do not appear in the client-propagated activity.

Instead, they appear in an activity correlated by transfer traces to the activity whose ID is propagated by the client.

Note:

If the propagateActivity attribute is set to true on both the client and service, the ambient activity in the operation scope of the service is set by WCF.

You can use the following code to check whether an activity was set in scope by WCF.

```
// Check if an activity was set in scope by WCF, i.e., if it was
// propagated from the client. If not, i.e., ambient activity is
// equal to Guid.Empty, create a new one.
if(Trace.CorrelationManager.ActivityId == Guid.Empty)
{
    Guid newGuid = Guid.NewGuid();
    Trace.CorrelationManager.ActivityId = newGuid;
}
// Emit your Start trace.
ts.TraceEvent(TraceEventTypes.Start, 0, "Add Activity");

// Emit the processing traces for that request.
serviceTs.TraceInformation("Service receives Add " + n1 + ", " + n2);

// double result = n1 + n2;
serviceTs.TraceInformation("Service sends Add result" + result);
```

```
// Emit the Stop trace and exit the method scope.
ts.TraceEvent(TraceEventType.Stop, 0, "Add Activity");
// return result;
```

Emitting User-Code Traces

(<http://msdn.microsoft.com/en-us/library/aa738759.aspx>)

Tracing and Message Logging

(<http://msdn.microsoft.com/en-us/library/ms751526.aspx>)

Propagation

(<http://msdn.microsoft.com/en-us/library/aa738751.aspx>)

Service Trace Viewer Tool (SvcTraceViewer.exe)

(<http://msdn.microsoft.com/en-us/library/ms732023.aspx>)

QUESTION 116

You are developing a Windows Communication Foundation (WCF) service. One of the service operations contains the following code.

```
private static int counter = 0;
[OperationContract]
public void IncrementCount()
{
    counter++;
}
```

You need to set a service behavior that prevents two or more threads from incrementing the counter variable at the same time.

Which code segment should you use to set the service behavior?

- A. [ServiceBehavior(InstanceContextMode = InstanceContextMode.Single, ConcurrencyMode = ConcurrencyMode.Single)]
- B. [ServiceBehavior(InstanceContextMode = InstanceContextMode.PerSession, ConcurrencyMode = ConcurrencyMode.Single)]
- C. [ServiceBehavior(InstanceContextMode = InstanceContextMode.Single, ConcurrencyMode = ConcurrencyMode.Multiple)]
- D. [ServiceBehavior(InstanceContextMode = InstanceContextMode.PerCall, ConcurrencyMode = ConcurrencyMode.Reentrant)]

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

"prevents two or more threads...at the same time" --> ConcurrencyMode.Single

ConcurrencyMode Enumeration

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.concurrencymode.aspx>)

InstanceContextMode Enumeration

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.instancecontextmode.aspx>)

QUESTION 117

You are developing a Windows Communication Foundation (WCF) service to provide shopping cart support.

ASP.NET compatibility mode is not enabled. The shopping cart information must be retained across user visits to the store until the user explicitly empties the cart or submits the cart contents to order.

You need to implement the service as a DurableService.

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

- A. Use basicHttpBinding for both the client application and the service.
- B. Create the persistence provider database and configure the persistenceProvider element of the service behavior to point to that database.
- C. Use wsHttpContextBinding for both the client application and the service.
- D. In the method to add an item to the shopping cart, serialize the shopping cart contents after adding the current item and storing it in a Session variable.

Correct Answer: BC

Section: (none)

Explanation

Explanation/Reference:

Durable services are Windows Communication Foundation (WCF) services that use context bindings, attributes, and new configuration elements to persist service state information to a store, such as a SQL database, which means that durable services have the capability to restore their own state when they are recycled. Durable services are useful when implementing long-running stateful conversations.

Durable services are Windows Communication Foundation (WCF) services that are designed to be durable and long-running, and to survive application and server restarts. Durable services are persisted using either the out-of-the-box provider created by a SqlPersistenceProviderFactory or a custom persistence provider derived from the abstract class PersistenceProvider or LockingPersistenceProvider.

Caution:

To use SqlPersistenceProviderFactory, create your persistence database using the scripts from .NET Framework 3.0.

These scripts install by default to the C:\Windows\Microsoft.NET\Framework\v3.0\Windows Workflow Foundation\SQL\EN*.sql directory.

Tutorial: Create a Durable Service

([http://msdn.microsoft.com/en-us/library/bb628514\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/bb628514(v=vs.90).aspx))

QUESTION 118

A Windows Communication Foundation (WCF) service is deployed with netTcpBinding.

This service uses a duplex message exchange pattern. You are developing the next version of the WCF service.

You discover that your company's hardware load balancer performs correctly only for WCF services that use HTTP.

You need to ensure that your service works with the load balancer. What should you do?

- A. Use basicHttpBinding.
- B. Create a custom binding that has the compositeDuplex, textMessageEncoding, and namedPipeTransport binding elements in this order.
- C. Create a custom binding that has the compositeDuplex, textMessageEncoding, and wsHttpTransport binding elements in this order.
- D. Use wsHttpBinding.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

Custom Bindings

You can add binding elements by adding the desired BindingElement objects to its Elements collection. The order in which the binding element is added is very important. The order of adding the binding elements is as follows:

1. Transaction Flow (not required)
2. Reliable Messaging (not required)
3. Message Security (not required)
4. Composite Duplex (not required)
5. Message Encoding (required)
6. Transport Security (not required)
7. Transport (required)

Bindings

(<http://msdn.microsoft.com/en-us/library/ff650874.aspx>)

WCF Bindings in Depth

(<http://msdn.microsoft.com/en-us/magazine/cc163394.aspx>)

QUESTION 119

You are developing a Windows Communication Foundation (WCF) service that executes a long-running operation.

The service is accessed from your business applications in a server transaction in which the client does not participate.

You need to ensure that the transaction times out and aborts if the operation has not completed within 45 seconds.

What should you do?

- A. Set the service binding `sendTimeout` attribute to 00:00:45.
- B. Apply `[ServiceBehavior(TransactionTimeout="00:00:45")]` to the service implementation.
- C. Set the service binding `receiveTimeout` attribute to 00:00:45.
- D. Apply `[OperationBehavior(TransactionScopeRequired=false)]` to the service operation.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

ServiceBehaviorAttribute.TransactionTimeout Property

Gets or sets the period within which a transaction must complete.

ServiceBehaviorAttribute.TransactionTimeout Property

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.servicebehaviorattribute.transactiontimeout.aspx>)

QUESTION 120

You are developing a Windows Service. The Windows Service will host a Windows Communication Foundation (WCF) service.

The Windows Service class will inherit from `ServiceBase`.

You need to ensure that the WCF service starts when the Windows Service is restarted. What should you do in the Windows Service class?

- A. - Create a public method named `Main`.
- Create a new `ServiceHost` in the `Main` method.
- Override the `OnShutdown` method and close the `ServiceHost`.
- B. - Override the `OnStart` method and create and open a new `ServiceHost`.
- Override the `OnStop` method and close the `ServiceHost`.
- C. - Override the `OnPowerEvent` method and open a new `ServiceHost`.
- Override the `OnShutdown` method and close the `ServiceHost`.
- D. - Override the `OnContinue` method and open a new `ServiceHost`.
- Override the `OnStop` method and close the `ServiceHost`.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

Windows services can be managed with the Microsoft.ManagementConsole.SnapIn in Microsoft Management Console (MMC)

and can be configured to start up automatically when the system boots up.

This hosting option consists of registering the application domain (AppDomain) that hosts a WCF service as a managed

Windows service so that the process lifetime of the service is controlled by the Service Control Manager (SCM) for Windows services.

The service code includes a service implementation of the service contract, a Windows Service class, and an installer class.

The service implementation class, CalculatorService, is a WCF service. The CalculatorWindowsService is a Windows service.

To qualify as a Windows service, the class inherits from ServiceBase and implements the OnStart and OnStop methods.

In OnStart, a ServiceHost is created for the CalculatorService type and opened. In OnStop, the service is stopped and disposed.

The host is also responsible for providing a base address to the service host, which has been configured in application settings.

The installer class, which inherits from Installer, allows the program to be installed as a Windows service by the Installutil.exe tool.

QUESTION 121

You are developing a Windows Communication Foundation (WCF) service to provide an in-memory cache for many Web applications.

The service contract is defined as follows. (Line numbers are included for reference only.)

```
01 [ServiceContract]
02 public interface IDataCache
03 {
04     ...
05 }
06
07
08 public class DataCache : IDataCache
09 {
10     ...
11 }
```

You need to ensure that all users share the cache. Which code segment should you insert at line 07?

- A. [ServiceBehavior(TransactionIsolationLevel = IsolationLevel.RepeatableRead)]
- B. [ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]
- C. [ServiceBehavior(InstanceContextMode = InstanceContextMode.PerSession)]
- D. [ServiceBehavior(TransactionIsolationLevel = IsolationLevel.ReadComitted)]

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

InstanceContextMode Enumeration

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.instancecontextmode.aspx>)

PerSession A new InstanceContext object is created for each session.

PerCall A new InstanceContext object is created prior to and recycled subsequent to each call. If the channel does not create a session this value behaves as if it were PerCall.

Single Only one InstanceContext object is used for all incoming calls and is not recycled subsequent to the calls. If a service object does not exist, one is created.

QUESTION 122

You develop a Windows Communication Foundation (WCF) service that uses basic authentication for client credentials.

This service is currently configured to use message security. The service is hosted on a server in workgroup mode.

Users report that their passwords are stolen when they use public computers.

You need to ensure that messages are secure and users are authenticated.

You prevent the service from being called over HTTP through Microsoft Internet Information Services (IIS) configuration.

What should you do next?

- A. Use the transport security mode and specify None for transport client credential type.
- B. Use the transportWithMessageCredential security mode and specify Basic for the transport client credential type.
- C. Use the message security mode and specify Basic for the transport client credential type.
- D. Use the transportWithMessageCredential security mode and specify None for the transport client credential type.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

By default, the wsHttpBinding binding provides HTTP communication. When configured for transport security, the binding supports HTTPS communication. HTTPS provides confidentiality and integrity protection for the messages that are transmitted over the wire. However the set of authentication mechanisms that can be used to authenticate the client to the service is limited to what the HTTPS transport supports. Windows Communication Foundation (WCF) offers a TransportWithMessageCredential security mode that is designed to overcome this limitation.

When this security mode is configured, the transport security is used to provide confidentiality and integrity for the transmitted messages and to perform the service authentication. However, the client authentication is performed by putting the client credential directly in the message. This allows you to use any credential type that is supported by the message security mode for the client authentication while keeping the performance benefit of transport security mode.

QUESTION 123

You are developing a Windows Communication Foundation (WCF) service. One of the parameters used with the

service operations is a security token. The security token is not sensitive.

The monitoring software tracks security tokens and can read tokens in clear text only.

The company security policy requires that you validate all clear text data passed over the corporate network.

You need to ensure that the service verifies that the security token is not changed during transit. What should you do?

- A. For all the security-sensitive members, set the ProtectionLevel parameter of the MessageBodyMember or MessageHeader attribute to EncryptAndSign.
- B. Implement IEndpointIdentityProvider in the message contract class.
- C. Implement ISecureConversationSession in the message contract class.
- D. For all the security-sensitive members, set the ProtectionLevel parameter of the MessageBodyMember or MessageHeader attribute to Sign.

Correct Answer: D

Section: (none)**Explanation****Explanation/Reference:**

"can read tokens in clear text only" + " verifies that ...is not changed during transit" --> ProtectionLevel.Sign

To understand the protection level feature, the following basic statements apply:

Three basic levels of protection exist for any part of a message. The property (wherever it occurs) is set to one of the ProtectionLevel enumeration values. In ascending order of protection, they include:

None.

Sign. The protected part is digitally signed. This ensures detection of any tampering with the protected message part.

EncryptAndSign. The message part is encrypted to ensure confidentiality before it is signed.

How to set ProtectionLevel Property

(<http://msdn.microsoft.com/en-us/library/aa347791.aspx>)

QUESTION 124

You are developing a Windows Communication Foundation (WCF) service.

The following code defines and implements the service. (Line numbers are included for reference only.)

```

01  [ServiceContract(SessionMode = SessionMode.Allowed)]
02  public interface ICatchAll
03  {
04      [OperationContract(IsOneWay = false, Action = "*", ReplyAction = "*")]
05      Message ProcessMessage(Message message);
06  }
07
08  public class CatchAllService : ICatchAll
09  {
10      public Message ProcessMessage(Message message)
11      {
12
13          ...
14          return returnMsg;
15      }
16  }

```

You need to ensure that two identical copies of the received message are created in the service.

Which code segment should you insert at line 12?

- A. `Message msgCopy = message.CreateBufferedCopy(8192) as Message;`
`Message returnMsg = message.CreateBufferedCopy(8192) as Message;`
- B. `MessageBuffer buffer = message.CreateBufferedCopy(8192);`
`Message msgCopy = buffer.CreateMessage();`
`Message returnMsg = buffer.CreateMessage();`
- C. `MessageBuffer buffer = message.CreateBufferedCopy(8192);`
`Message msgCopy = buffer.CreateMessage();`
`Message returnMsg = msgCopy;`
- D. `Message msgCopy = message;`
`Message returnMsg = message;`

Correct Answer: B

Section: (none)

Explanation**Explanation/Reference:****MessageBuffer Class**

Represents a memory buffer that stores an entire message for future consumption.

MessageBuffer.CreateMessage Method

Returns a copy of the original message.

MessageBuffer Class

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.channels.messagebuffer.aspx>)

QUESTION 125

You are developing a Windows Communication Foundation (WCF) REST service to provide access to a library book catalog.

The following code segment defines the service contract. (Line numbers are included for reference only.)

```
01 [ServiceContract()]
02 [AspNetCompatibilityRequirements(RequirementsMode =
AspNetCompatibilityRequirementsMode.Allowed)]
03 public Class LibraryService
04 {
05     public Book GetBookByTitle(string title)
06     {
07         ...
08     }
09
10     [WebGet(UriTemplate = "Book/{id}")]
11     public Book GetBookById(string id)
12     {
13         ...
14     }
15 }
```

Library patrons want the ability to search the catalog by title.

You need to ensure that the GetBookByTitle method is exposed as a service method.

Which code segment should you insert at line 04?

- A. [WebGet(UriTemplate = "Book/{title}")]
- B. [WebGet(UriTemplate = "BookByTitle/{title}")]
- C. [WebGet(UriTemplate = "Book/{titleToSearch}")]
- D. [WebGet(UriTemplate = "{titleToSearch}")]

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

AspNetCompatibilityRequirementsAttribute Class

Applied to a service to indicate whether that service can be run in ASP.NET compatibility code.

AspNetCompatibilityRequirementsMode Enumeration

Specifies whether a service runs, or can run, in a mode that is compatible with ASP.NET.

NotAllowed services must run in an application domain with ASP.NET compatibility mode set to false.

Allowed services can run in an application domain with ASP.NET compatibility mode set to true or false.

Required services must run in an application domain with ASP.NET compatibility mode set to true.

AspNetCompatibilityRequirementsMode Enumeration

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.activation.aspNetcompatibilityrequirementsmode.aspx>)

QUESTION 126

You are creating an application using Visual Studio 2010. The application consumes a Windows Communication Foundation (WCF) service.

You are adding a service reference to the WCF service. You need to ensure that the generated proxy does not block

the calling thread when executing a service method. What should you do when adding the service reference?

- A. Set the Collection type to ObservableCollection.
- B. Select the Reuse types in all referenced assemblies option.
- C. Select the Generate asynchronous operations option.
- D. Select the Always generate message contracts option.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

The Configure Service Reference dialog box enables you to configure the behavior of Windows Communication Foundation (WCF) services.

To access the Configure Service Reference dialog box, right-click a service reference in Solution Explorer and choose Configure Service Reference.

You can also access the dialog box by clicking the Advanced button in the Add Service Reference Dialog Box.

To change the address where a WCF service is hosted, enter the new address in the Address field.

To change the access level for classes in a WCF client, select an access-level keyword in the Access level for generated classes list.

To call the methods of a WCF service asynchronously, select the Generate asynchronous operations check box.

To generate message contract types in a WCF client, select the Always generate message contracts check box.

To specify list or dictionary collection types for a WCF client, select the types from the Collection type and Dictionary collection type lists.

To disable type sharing, clear the Reuse types in referenced assemblies check box. To enable type sharing for a subset of referenced assemblies,

select the Reuse types in referenced assemblies check box, select Reuse types in specified referenced assemblies, and select the desired references in the Referenced assemblies list.

QUESTION 127

You are developing a Windows Communication Foundation (WCF) service that is used to check the status of orders placed by customers.

The following code segment is part of your service.

(Line numbers are included for reference only.)

```
01 [ServiceContract]
02 public interface IStatus
03 {
04     [OperationContract]
05     int GetOrderStatus(string orderNumber);
06 }
07
08 class OrderService : IStatus
09 {
10     public int GetOrderStatus(string orderNumber)
11     {
12         ...
13     }
14 }
15
16 class Program
17 {
18     static void Main(string[] args)
19     {
20
21         host.Open();
22         ...
23     }
24 }
25 }
```

You need to ensure that the service always listens at net.pipe://SupplyChainServer/Pipe. What should you do?

- A. Insert the following code at line 20.

```
ServiceHost host = new ServiceHost(typeof(IStatus));
```

Insert the following code at 21.

```
host.AddServiceEndpoint(typeof(IStatus),  
    new NetTcpBinding(),  
    "net.pipe://SupplyChainServer/Pipe");
```

- B. Insert the following code at line 20.

```
ServiceHost host = new ServiceHost(typeof(OrderService));
```

Insert the following code at 21.

```
host.AddServiceEndpoint(typeof(IStatus),  
    new NetNamedPipeBinding(),  
    "net.pipe://SupplyChainServer/Pipe");
```

- C. Insert the following code at line 20.

```
ServiceHost host = new ServiceHost(typeof(IStatus));
```

Insert the following code at 21.

```
host.AddServiceEndpoint(typeof(OrderService),  
    new NetTcpBinding(),  
    "net.pipe://SupplyChainServer/Pipe");
```

- D. Insert the following code at line 20.

```
ServiceHost host = new ServiceHost(typeof(OrderService));
```

Insert the following code at 21.

```
host.AddServiceEndpoint(typeof(OrderService),  
    new NetNamedPipeBinding(),  
    "net.pipe://SupplyChainServer/Pipe");
```

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 128

You are developing a custom service host for a Windows Communication Foundation (WCF) service. The service host is named **MovieServiceHost**.

You need to deploy the service with the custom service host in Microsoft Internet Information Services (IIS) 7.0.

What should you do?

- A. Make sure that service class has a default constructor. Add a public read-only property with the name **ServiceHost** that returns an instance of the **MovieServiceHost** class.
- B. Create a factory for the custom service host. Name the factory **MovieServiceHostFactory**. In the web.config file, add the following attribute to the **<add>** element within the **<serviceActivations>** element. `factory="MovieServiceFactory"`
- C. Create a factory for the custom service host. Name the factory **MovieServiceHostFactory**. In the .svc file, add the following line. `<%@ServiceHost Service="MovieServiceHostFactory" Language="C#">`
- D. Decorate the custom service host class with the following line.
`[System.ServiceModel.Activation.ServiceActivationBuildProvider()]`

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 129

You are developing a Windows Communication Foundation (WCF) service that will be hosted in Microsoft Internet Services (IIS) 7.0.

The service must be hosted in an IIS application named **Info**. You need to enable this service to be hosted in IIS by changing the **web.config** file.

Which XML segment should you add to the **web.config** file?

- A.

```
<serviceHostingEnvironment>
  <serviceActivations>
    <add relativeAddress="Info" service="Info.svc" />
  </serviceActivations>
</serviceHostingEnvironment>
```
- B.

```
<serviceHostingEnvironment>
  <transportConfigurationTypes>
    <add name="Info.svc" transportConfigurationType="FileNotRequired" />
  </transportConfigurationTypes>
</serviceHostingEnvironment>
```
- C.

```
<serviceHostingEnvironment>
  <serviceActivations>
    <add relativeAddress="Info.svc" service="Info" />
  </serviceActivations>
</serviceHostingEnvironment>
```
- D.

```
<serviceHostingEnvironment>
  <transportConfigurationTypes>
    <add name="Info" transportConfigurationType="Info.svc" />
  </transportConfigurationTypes>
</serviceHostingEnvironment>
```

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 130

You are developing a Windows Communication Foundation (WCF) service. You enable logging in the configuration file.

The opening tag is defined as follows.

```
<messageLogging logEntireMessage="true"
  logMalformedMessages="true"
  logMessagesAtServiceLevel="true"
  logMessagesAtTransportLevel="true"
  maxMessagesToLog="20">
```

You need to ensure that logging is implemented so that only messages with SOAP headers are logged. What should you add to the filters element of the application configuration file?

- A.

```
<add xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  /Actions[starts-with(text(), 'soap:Envelope')]
</add>
```
- B.

```
<add xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  soap:Header
</add>
```
- C.

```
<add xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  /soap:Envelope/soap:Header
</add>
```
- D.

```
<add xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  /Actions[starts-with(text(), 'soap:Header')]
</add>
```

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 131

You are developing a Windows Communication Foundation (WCF) service named **CalculatorService**, which implements the **ICalculatorService** contract. The service is configured to be discoverable through UDP. **CalculationService** contains multiple endpoints. One of the endpoints is configured with the following behavior.

```
<behavior name="calculatorEndpointBehavior">
  <endpointDiscovery enabled="true">
    <extentions>
      <Information>
        ICalculationService Endpoint.
      </Information>
      <Information>
        Udp Exposed Calculator Endpoint
      </Information>
    </extentions>
  </endpointDiscovery>
</behavior>
```

You need to log all the endpoint metadata information that is added by the service host. Which code segment should you use?

- A.

```
var discoveryClient =
    new DiscoveryClient(new UdpDiscoveryEndpoint());
var findCriteria =
    new FindCriteria(typeof(ICalculatorService));
var findResponse = discoveryClient.Find(findCriteria);
var meta = findResponse.Endpoints[0];

foreach(var xElement in meta.Extentions)
{
    Log("Endpoint Information: "
        + xElement.Element("Information").Value);
}
```
- B.

```
var discoveryClient =
    new DiscoveryClient(new UdpDiscoveryEndpoint());
var findCriteria = new FindCriteria();
var findResponse = discoveryClient.Find(findCriteria);
var meta = discoveryClient.Endpoint;

foreach(var xElement in meta.Contract.Operations)
{
    Log("Endpoint Information: "
        + xElement.Behaviors.ToString());
}
```
- C.

```
var discoveryClient =
    new DiscoveryClient(new UdpDiscoveryEndpoint());
var findCriteria = new FindCriteria();
var findResponse = discoveryClient.Find(findCriteria);

foreach(var meta in findResponse.Endpoints)
{
    foreach(var xElement in meta.Extentions)
    {
        Log("Endpoint Information: "
            + xElement.Element("Information").Value);
    }
}
```
- D.

```
var discoveryClient =
    new DiscoveryClient(new UdpDiscoveryEndpoint());
```

```

var findCriteria =
    new FindCriteria(typeof(ICalculatorService));
var findResponse = discoveryClient.Find(findCriteria);

foreach(var meta in findResponse.Endpoints)
{
    foreach(var xElement in meta.Extentions)
    {
        Log("Endpoint Information: "
            + xElement.Element("Information").Value);
    }
}

```

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 132

A Windows Communication Foundation (WCF) service listens for messages at `net.tcp://www.contoso.com/MyService`.

It has a logical address at `http://www.contoso.com/MyService`. The configuration for the WCF client is as follows:

```

<endpoint address="http://www.contoso.com/MyService"
    binding="netTcpBinding"
    bindingConfiguraton="NetTcpBinding_IMyService"
    contract="ServiceReferencel.IMyService"
    name="NetTcpBinding_IMyService"/>

```

The generated configuration does not provide enough information for the client to communicate with the server.

You need to update the client so that it can communicate with the server. What should you do?

- A. In the client configuration, change the value of the address attribute to `net.tcp://www.contoso.com/MyService`
- B. In the client configuration, change the value of the address attribute to `net.tcp://www.contoso.com/MyService` `listen=http://www.contoso.com/MyService`.
- C. After instantiating the client and before invoking any service operation, add this line of code.
`EndpointBehaviors.Add(new EndpointDiscoveryBehavior(){ Enabled = true });`
- D. After instantiating the client and before invoking any service operation, add this line of code.
`client.Endpoint.Behaviors.Add(new ClientViaBehavior(new Uri("net.tcp://www.contoso.com/IMyService")));`

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

The **ClientViaBehavior** is used by clients to specify the URI for which the transport channel should be created.

If such a behavior exists in the behavior collection on a client endpoint, the transport uses the URI that it provides, while all the other channel layers in the stack use the `EndpointAddress` provided at `ChannelFactory` construction time.

This `EndpointAddress` also becomes the WS-Addressing To header.

Use this behavior when the immediate network destination is not the intended processor of the message to enable multiple-hop conversations when the calling application does not necessarily know the ultimate destination or when the destination Via header is not an address.

You can also set this value using the <clientVia> element in a client application configuration file.

ClientViaBehavior Class

(<http://msdn.microsoft.com/en-us/library/system.servicemodel.description.clientviabehavior.aspx>)

QUESTION 133

You are developing a Windows Communication Foundation (WCF) service that allows customers to update financial data.

The service contract is defined as follows. (Line numbers are included for reference only)

```
01 [ServiceContract]
02 public interface IDataUpdate
03 {
04     [OperationContract]
05     [TransactionFlow(TransactionFlowOption.Mandatory)]
06     void Update(string accountNumber, double amount);
07 }
08
09 class UpdateService : IDataUpdate
10 {
11     [OperationBehavior(TransactionScopeRequired = true,
TransactionAutoComplete = false)]
12     public void Update(string accountNumber, double amount)
13     {
14         ...
15     }
16 }
17
```

You need to ensure that the service is invoked within the transaction. What should you do?

- A. Replace line 01 with the following code
[ServiceContract(SessionMode = SessionMode.NotAllowed)]
- B. Replace line 01 with the following code
[ServiceContract(SessionMode = SessionMode.Required)]
- C. Call the Invoke method of the form and supply a delegate.
- D. Call the BeginInvoke method of the form and supply a delegate.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

The contract for the service defines that all of the operations require a transaction to be flowed with requests:

```
[ServiceContract(Namespace = "http://Microsoft.ServiceModel.Samples", SessionMode =
SessionMode.Required)]
public interface ICalculator
{
    [OperationContract]
    [TransactionFlow(TransactionFlowOption.Mandatory)]
    double Add(double n);
    [OperationContract]
    [TransactionFlow(TransactionFlowOption.Mandatory)]
```

```

double Subtract(double n);
[OperationContract]
[TransactionFlow(TransactionFlowOption.Mandatory)]
double Multiply(double n);
[OperationContract]
[TransactionFlow(TransactionFlowOption.Mandatory)]
double Divide(double n);
}

```

Service Transaction Behavior

(<http://msdn.microsoft.com/en-us/library/ms751413.aspx>)

QUESTION 134

A client application calls a Windows Communication Foundation (WCF) service with a proxy class that was generated by Visual Studio.

The service binding was changed from wsHttpBinding to basicHttpBinding. The method signatures in the service contract are changed.

After these changes, the client application receives errors when calling methods on the service.

You need to ensure that the client can invoke methods on the service successfully. What should you do?

- A. Update the configuration file to use basicHttpBinding.
- B. Enable anonymous authentication for the service.
- C. Copy the configuration file elements under the <system.serviceModel> element from the service back to the client.
- D. Update the service reference.

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 135

You are developing a Windows Communication Foundation (WCF) service that will be hosted in Microsoft Internet Services (IIS) 7.0.

The service must be hosted in an IIS application named **Info**. You need to enable this service to be hosted in IIS by changing the **web.config** file.

Which XML segment should you add to the **web.config** file?

- A.

```
<serviceHostingEnvironment>
  <serviceActivations>
    <add relativeAddress="Info" service="Info.svc" />
  </serviceActivations>
</serviceHostingEnvironment>
```
- B.

```
<serviceHostingEnvironment>
  <transportConfigurationTypes>
    <add name="Info.svc" transportConfigurationType="FileNotRequired" />
  </transportConfigurationTypes>
</serviceHostingEnvironment>
```
- C.

```
<serviceHostingEnvironment>
  <serviceActivations>
    <add relativeAddress="Info.svc" service="Info" />
  </serviceActivations>
</serviceHostingEnvironment>
```
- D.

```
<serviceHostingEnvironment>
  <transportConfigurationTypes>
    <add name="Info" transportConfigurationType="Info.svc" />
  </transportConfigurationTypes>
</serviceHostingEnvironment>
```

Correct Answer: C

Section: (none)**Explanation****Explanation/Reference:****QUESTION 136**

A Windows Communication Foundation (WCF) service implements a contract with one-way and request-reply operations.

The service is exposed over a TCP transport. Clients use a router to communicate with the service.

The router is implemented as follows. (Line numbers are included for reference only.)

```
01 ServiceHost host = new ServiceHost(typeof(RoutingService));
02 host.AddServiceEndpoint(
03     typeof(ISimplexDatagramRouter),
04     new NetTcpBinding(), "net.tcp://localhost/Router"
05 );
06 List<ServiceEndpoint> lep = new List<ServiceEndpoint>();
07 lep.Add(
08     new ServiceEndpoint(
09         ContractDescription.GetContract(
10             typeof(ISimplexDatagramRouter)
11         ),
12         new NetTcpBinding(),
13         new EndpointAddress("net.tcp://localhost:8080/Logger")
14     );
15 );
16 RoutingConfiguration rc = new RoutingConfiguration();
17 rc.FilterTable.Add(new MatchAllMessageFilter(), lep);
18 host.Description.Behaviors.Add(new RoutingBehavior(rc));
```

Request-reply operations are failing. You need to ensure that the router can handle one-way and request-reply operations.

What should you do?

- A. Change line 03 as follows:
typeof(IRequestReplyRouter),
- B. Change line 03 as follows:
typeof(IDuplexSessionRouter),
- C. Change line 10 as follows:
typeof(IRequestReplyRouter)
- D. Change line 10 as follows:
typeof(IDuplexSessionRouter)

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

IDuplexSessionRouter Interface Defines the interface required to process messages from duplex session channels.

IDuplexSessionRouter Interface

([http://msdn.microsoft.com/en-us/library/system.servicemodel.routing.iduplexsessionrouter\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/system.servicemodel.routing.iduplexsessionrouter(v=vs.110).aspx))

QUESTION 137

You develop a Windows Communication Foundation (WCF) service. You enable all performance counters and run multiple calls to the service.

The service must isolate session data to each user. You need to monitor the instancing behaviour used in the service.

Which performance counter should you monitor?

- A. ServiceModeService 4.0.0.0\Calls
- B. ServiceModeService 4.0.0.0\Instances
- C. ASP.Net StateService \ Server Sessions Active
- D. ASP.Net StateService \ Server Sessions Total

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

"monitor the instancing behaviour" --> ServiceModeService 4.0.0.0\Instances

QUESTION 138

You develop a Windows Communication Foundation (WCF) service that uses basic authentication for client credentials.

This service is currently configured to use message security. The service is hosted on a server in workgroup mode.

Users report that their passwords are stolen when they use public computers.

You need to ensure that messages are secure and users are authenticated.

You prevent the service from being called over HTTP through Microsoft Internet Information Services (IIS) configuration.

What should you do next?

- A. Use the transport security mode and specify None for transport client credential type.
- B. Use the transportWithMessageCredential security mode and specify Basic for the transport client credential type.
- C. Use the message security mode and specify Basic for the transport client credential type.
- D. Use the transportWithMessageCredential security mode and specify None for the transport client credential type.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

By default, the wsHttpBinding binding provides HTTP communication. When configured for transport security,

the binding supports HTTPS communication. HTTPS provides confidentiality and integrity protection for the messages

that are transmitted over the wire. However the set of authentication mechanisms that can be used to authenticate

the client to the service is limited to what the HTTPS transport supports. Windows Communication Foundation (WCF)

offers a TransportWithMessageCredential security mode that is designed to overcome this limitation.

When this security mode is configured, the transport security is used to provide confidentiality and integrity for the transmitted

messages and to perform the service authentication. However, the client authentication is performed by putting the client credential

directly in the message. This allows you to use any credential type that is supported by the message security mode for the client

authentication while keeping the performance benefit of transport security mode.



<http://www.gratisexam.com/>