

Microsoft 70-503

**TS: Microsoft .NET Framework 3.5 - Windows
Communication Foundation
Version: 12.0**

Topic 1, C#**QUESTION NO: 1**

You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5. You have successfully defined a service contract named `IManageOrders`. You write the following code segment.

```
public class OrderImpl : IManageOrders {
    public void MarkOrderClosed(int orderId) {
        try {
            ...
        }
        catch (SqlException exc) {
            throw new FaultException<DataFault>(new DataFault());
        }
    }
}

[DataContract]
public class DataFault {
}
```

You need to create a fault contract for the `MarkOrderClosed` method on the `IManageOrders` service contract. Which code segment should you add?

- A. `[FaultContract(typeof(DataFault))]`
- B. `[FaultContract(typeof(Exception))]`
- C. `[FaultContract(typeof(SqlException))]`
- D. `[FaultContract(typeof(FaultException))]`

Answer: A

Explanation:

QUESTION NO: 2

You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5. You have successfully created two interfaces: `IMyService` and `IMyServiceClient`.

You need to ensure that the service is able to call methods from the client application by using the `IMyServiceClient` interface.

Which code segment should you use?

- A. `[ServiceContract(CallbackContract=typeof(IMyServiceClient))]`
`public interface IMyService`
`{`
 `...`
`}`
`public interface IMyServiceClient`
`{`
 `...`
`}`
- B. `[ServiceContract(CallbackContract=typeof(IMyService))]`
`public interface IMyService`
`{`
 `...`
`}`
`public interface IMyServiceClient`
`{`
 `...`
`}`
- C. `[ServiceContract(SessionMode=SessionMode.Allowed)]`
`[ServiceKnownType(typeof(IMyServiceClient))]`
`public interface IMyService : IMyServiceClient`
`{`
 `...`
`}`
`public interface IMyServiceClient`
`{`
 `...`
`}`
- D. `[ServiceContract]`
`[ServiceKnownType(typeof(IMyServiceClient))]`
`public interface IMyService`
`{`
 `...`
`}`
`[ServiceContract]`
`public interface IMyServiceClient : ICommunicationObject`
`{`
 `...`
`}`

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

Answer: A

Explanation:

QUESTION NO: 3

You are replacing an ASMX Web service with a Windows Communication Foundation service by using Microsoft .NET Framework 3.5. The ASMX Web service has a number of existing clients, which you cannot recompile and redeploy. You need to maintain backward compatibility with the existing clients. Which code fragment should you use?

- A. [ServiceContract]
public interface IQuoteService {
 [OperationContract]
 Quote GetQuote(string tickerSymbol);
}
[DataContract]
public class Quote : ISerializable {
 ...
}
- B. [ServiceContract]
public interface IQuoteService {
 [OperationContract]
 Quote GetQuote(string tickerSymbol);
}
[DataContract, XmlSerializerFormat]
public class Quote {
 ...
}
- C. [ServiceContract, XmlSerializerFormat]
public interface IQuoteService {
 [OperationContract]
 Quote GetQuote(string tickerSymbol);
}
public class Quote {
 ...
}
- D. [ServiceContract]
public interface IQuoteService {
 [OperationContract]
 Quote GetQuote(string tickerSymbol);
}
[DataContract]
public class Quote {
 ...
}
- E. [ServiceContract]
public interface IQuoteService {
 [OperationContract]
 Quote GetQuote(string tickerSymbol);
}

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

Answer: B

Explanation: XMLSerializer

We can find XMLSerializer in the System.Xml.Serialization namespace. WCF supports this serialization from .Net 1.0. It is used by default with the ASP.Net webservices (ASMX).

Usage

We can use this serializer whenever we want to get backward compatibility with ASMX. It can also be used when integrating with non WCF Services.

QUESTION NO: 4

You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5. The service has a contract named IMyService.

You need to ensure that the service uses an XMLSerializer object.

Which code segment should you use?

- A. `[ServiceContract]`
`[XmlSerializerFormat]`
`public interface IMyService`
`{`

`}`
- B. `[ServiceContract]`
`[XmlSerializerAssembly]`
`public interface IMyService`
`{`

`}`
- C. `[ServiceContract]`
`[XmlType]`
`public interface IMyService`
`{`

`}`
- D. `[ServiceContract]`
`[DataContractFormat (Style=OperationFormatStyle.Document)]`
`public interface IMyService`
`{`

`}`

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

Answer: A

Explanation: XmlSerializerFormatAttribute Class

Instructs the Windows Communication Foundation (WCF) infrastructure to use the XmlSerializer instead of the XmlObjectSerializer.

Note: The following example applies the XmlSerializerFormatAttribute to an interface.

```
[ServiceContract, XmlSerializerFormat (Style = OperationFormatStyle.Rpc,  
Use = OperationFormatUse.Encoded)]  
public interface ICalculator  
{  
[OperationContract, XmlSerializerFormat (Style = OperationFormatStyle.Rpc,  
Use = OperationFormatUse.Encoded)]  
double Add(double a, double b);  
}
```

Reference: XmlSerializerFormatAttribute Class

QUESTION NO: 5

You have created a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

The existing service interface is named `IMyService`, and contains the following code segment.

```
[ServiceContract (Name="SvcOrder",  
    Namespace="http://contoso.com/services")]  
public interface IMyService  
{  
    [OperationContract]  
    void DoSomething();  
}
```

You create a new service named `IMyServiceV1` that contains an operation named `DoSomethingElse`.

You need to ensure that existing client applications are still able to access the `IMyService.DoSomething` method without modifying client code.

Which code segment should you use?

- A. `[ServiceContract (Name="SvcOrder",
Namespace="http: //contoso.com/services")]`
`public interface IMyServiceV1 : IMyService`
`{`
`[OperationContract]`
`void DoSomethingElse();`
`}`
- B. `[ServiceContract (Name="SvcOrder")]`
`public interface IMyServiceV1 : IMyService`
`{`
`[OperationContract]`
`void DoSomethingElse();`
`}`
- C. `[ServiceContract (Name="SvcOrderV1",
Namespace="http: //contoso.com/services")]`
`public interface IMyServiceV1 : IMyService`
`{`
`[OperationContract]`
`void DoSomethingElse();`
`}`
- D. `[ServiceContract (Namespace="http: //contoso.com/services/V1")]`
`public interface IMyServiceV1 : IMyService`
`{`
`[OperationContract]`
`void DoSomethingElse();`
`}`

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

Answer: A

Explanation:

QUESTION NO: 6

You create a Windows Communication Foundation service by using Microsoft .NET Framework 3.5. The service contains the following code segment.

```
[DataContract]
public class Person
{
    ...
}
```

```
[DataContract]
public class Customer : Person
{
    ...
}
```

You need to create a service contract that meets the following requirements:

- The service contract must have an operation contract named GetPerson that returns an object of Type Person.
- The GetPerson operation must be able to return an object of Type Customer.

Which code segment should you use?