

Microsoft 70-569

**70-569 UPGRADE: Transition your MCPD Enterprise
Application Developer Skills to MCPD Enterprise
Application Developer 3.5, Part 2**

Practice Test

Version 1.1

QUESTION NO: 1

You are creating a Windows Communication Foundation service by using Microsoft .NET Framework

3.5.

You write the following code segment.

```
namespace MyServices
{
[ServiceContract()]
interface IManageOrders
{
...
}
}
```

The service metadata must be exposed at the relative address named meta.

You need to add an endpoint element to the app.config file of the service host.

Which code fragment should you add

- A. `<endpoint address="meta" binding="wsHttpBinding" contract="IManageOrders" />`
- B. `<endpoint address="meta" binding="wsHttpBinding" contract="MyServices.IMetadataExchange" />`
- C. `<endpoint address="meta" binding="mexHttpBinding" contract="IMetadataExchange" />`
- D. `<endpoint address="meta" binding="mexHttpBinding" contract="MyServices.IManageOrders" />`

Answer: C

QUESTION NO: 2

You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

You write the following code segment.

```
Namespace MyServices
<ServiceContract()>Interface IManageOrders
...
End Interface
End Namespace
```

The service metadata must be exposed at the relative address named meta.
You need to add an endpoint element to the app.config file of the service host.

Which code fragment should you add

- A. `<endpoint address="meta" binding="wsHttpBinding" _
contract="IMangeOrders" />`
- B. `<endpoint address="meta" binding="wsHttpBinding" _
contract="MyServices.IMetadataExchange" />`
- C. `<endpoint address="meta" binding="mexHttpBinding" _
contract="IMetadataExchange" />`
- D. `<endpoint address="meta" binding="mexHttpBinding" _
contract="MyServices.IMangeOrders" />`

Answer: C

QUESTION NO: 3

You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

You have successfully defined a service contract named IMangeOrders.

You write the following code segment.

```
public class OrderImpl : IMangeOrders {  
    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 IMangeOrders 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

QUESTION NO: 4

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
    Implements IManageOrders
    Public Sub MarkOrderClosed(ByVal orderId As Integer) _
        Implements IManageOrders.MarkOrderClosed
    Try
    ...
    Catch ex As SqlException
    Throw New FaultException(Of DataFault)( _
        New DataFault())
    End Try
    End Sub
    End Class
    <DataContract()> _
    Public Class DataFault
    End Class
```

You need to create a fault contract for the MarkOrderClosed method on the IManageOrders service contract.

Which code segment should you add

- A. <FaultContract(GetType(DataFault))>
- B. <FaultContract(GetType(Exception))>
- C. <FaultContract(GetType(SqlException))>
- D. <FaultContract(GetType(FaultException))>

Answer: A

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(Namespace="http://contoso.com/services/V1")]`

```
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(Name="SvcOrder",
Namespace="http://contoso.com/services")]`

```
public interface IMyServiceV1 : IMyService
{
[OperationContract]
void DoSomethingElse();
}
```

Answer: D

QUESTION NO: 6

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()> _
Sub DoSomething()
End Interface
```

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(Namespace:="http://contoso.com/services/V1")> _
Public Interface IMyServiceV1
Inherits IMyService
<OperationContract()> _
Sub DoSomethingElse()
End Interface
- B. <ServiceContract(Name:="SvcOrder")> _
Public Interface IMyServiceV1
Inherits IMyService
<OperationContract()> _
Sub DoSomethingElse()
End Interface

C. `<ServiceContract(Name:="SvcOrderV1", _
Namespace:="http://contoso.com/services")> _
Public Interface IMyServiceV1
Inherits IMyService
<OperationContract()> _
Sub DoSomethingElse()
End Interface`

D. `<ServiceContract(Name:="SvcOrder", _
Namespace:="http://contoso.com/services")> _
Public Interface IMyServiceV1
Inherits IMyService
<OperationContract()> _
Sub DoSomethingElse()
End Interface`

Answer: D

QUESTION NO: 7

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

- A. [ServiceContract]
[ServiceKnownType("GetPerson")]
public interface IMyService
{
[OperationContract]
Person GetPerson();
}
- B. [ServiceContract]
public interface IMyService
{
[OperationContract]
[ServiceKnownType("Customer")]
Person GetPerson();
}
- C. [ServiceContract]
[ServiceKnownType(typeof(Customer))]
public interface IMyService
{
[OperationContract]
Person GetPerson();
}
- D. [ServiceContract]
[ServiceKnownType("GetPerson",typeof(Customer))]
public interface IMyService
{
[OperationContract]
Person GetPerson();
}

Answer: C

QUESTION NO: 8

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  
...  
End Class  
</DataContract(> _
```



```
Public Class Customer
Inherits Person
...
End Class
```

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

- A. `<ServiceContract()> _`
`<ServiceKnownType("GetPerson")> _`
Public Interface IMyService
`<OperationContract()> _`
Function GetPerson() As Person
End Interface
- B. `<ServiceContract()> _`
Public Interface IMyService
`<OperationContract()> _`
`<ServiceKnownType("Customer")> _`
Function GetPerson() As Person
End Interface
- C. `<ServiceContract()> _`
`<ServiceKnownType(GetType(Customer))> _`
Public Interface IMyService
`<OperationContract()> _`
Function GetPerson() As Person
End Interface
- D. `<ServiceContract()> _`
`<ServiceKnownType("GetPerson", GetType(Customer))> _`
Public Interface IMyService
`<OperationContract()> _`
Function GetPerson() As Person
End Interface

Answer: C

QUESTION NO: 9

You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

The service will contain an enumeration named OrderState.

The OrderState enumeration will contain the following four values:

Processing

Cancelled

Confirmed

Closed

The client application must be able to set the state of an Order entity to only the following two values:

Cancelled

Closed

You need to create the data contract for OrderState.

Which code segment should you use

A. [DataContract]

```
public enum OrderState
```

```
{
```

```
    Processing=1,
```

```
    [DataMember]
```

```
    Cancelled=2,
```

```
    [DataMember]
```

```
    Confirmed=3,
```

```
    Closed=4
```

```
}
```

B. [DataContract]

```
public enum OrderState
```

```
{
```

```
    Processing=1,
```

```
    [EnumMember]
```

```
    Cancelled=2,
```

```
    Confirmed=3,
```

```
    [EnumMember]
```

```
    Closed=4
```

```
}
```

C. [DataContract]

```
public enum OrderState
```

```
{
```

```
    [EnumMember(Value="False")]
```

```
    Processing=1,
```