



70-503

**TS: Microsoft .NET Framework 3.5 Windows Communication
Foundation Application Development**

Q&A

DEMO Version

Copyright (c) 2009 Chinatag LLC. All rights reserved.

Important Note Please Read Carefully

For demonstration purpose only, this free version Chinatag study guide contains **10** full length questions selected from our full version products which have more than **200** questions each.

This Study guide has been carefully written and compiled by Chinatag certification experts. It is designed to help you learn the concepts behind the questions rather than be a strict memorization tool. Repeated readings will increase your comprehension.

For promotion purposes, all PDF files are **not** encrypted. Feel free to distribute copies among your friends and let them know Chinatag website.

Study Tips

This product will provide you questions and answers along with detailed explanations carefully compiled and written by our experts. Try to understand the concepts behind the questions instead of cramming the questions. Go through the entire document at least twice so that you make sure that you are not missing anything.

Latest Version

We are constantly reviewing our products. New material is added and old material is revised. Free updates are available for 90 days after the purchase. You should check the products page on the <http://www.chinatag.com> website for an update 3-4 days before the scheduled exam date.

Please tell us what you think of our products. We appreciate both positive and critical comments as your feedback helps us improve future versions. Feedback on specific questions should be send to feedback@chinatag.com.

Thanks for purchasing our products and look forward to supplying you with all your Certification training needs.

Good studying!

Technical and Support Team
Chinatag LLC.

QUESTION 1

You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5. The service uses the net.tcp transport. You need to ensure that when the server starts, the service starts and continues to run. What should you do?

- A. Host the service in a Windows service.
- B. Host the service in a Windows Presentation Foundation application.
- C. Host the service under IIS 7.0 by using IIS 6.0 compatibility mode.
- D. Host the service under IIS 7.0 by using Windows Activation Services.

Answer: A

QUESTION 2

You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5. The service will be hosted in a managed Console application.

You want to add endpoints to the service. You need to ensure that all endpoints use the same base address. Which code fragment should you use?

- A.

```
[ServiceContract]
public interface IMortgageService {}
public class MortgageService : IMortgageService {}
Uri baseAddress=new Uri("http://localhost:8888/MortgageService");
ServiceHost serviceHost= new ServiceHost(typeof(MortgageService), new Uri[] {baseAddress});
serviceHost.AddServiceEndpoint(typeof(IMortgageService), new BasicHttpBinding(), "");
serviceHost.Open();
```
- B.

```
[ServiceContract]
public interface IMortgageService {}
public class MortgageService : IMortgageService {}
Uri baseAddress=new Uri("http://localhost:8888/MortgageService");
ServiceHost serviceHost= new ServiceHost(typeof(MortgageService), new Uri[]{});
serviceHost.AddServiceEndpoint(typeof(IMortgageService), new BasicHttpBinding(), baseAddress);
serviceHost.Open();
```
- C.

```
[ServiceContract]
public interface IMortgageService {}
public class MortgageService : IMortgageService {}
string baseAddress="http://localhost:8888/MortgageService";
ServiceHost serviceHost= new ServiceHost(typeof(MortgageService), new Uri[] {2 http://www.certmagic.com});
serviceHost.AddServiceEndpoint(typeof(IMortgageService), new BasicHttpBinding(), baseAddress);
serviceHost.Open();
```
- D.

```
[ServiceContract(Namespace="http://localhost:8888/MortgageService")]
public interface IMortgageService {}
public class MortgageService : IMortgageService {}
ServiceHost serviceHost = new ServiceHost(typeof(MortgageService), new Uri[] {});
serviceHost.AddServiceEndpoint(typeof(IMortgageService), new BasicHttpBinding(), "");
serviceHost.Open();
```

Answer: A

QUESTION 3

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

3.5.

You need to host the WCF service on the IIS Web server.

First, you create a new folder for your application files.

Next, you use the IIS management tool to create a Web application in the new folder.

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

- A. Create a web.config file that contains the appropriate configuration code.
Place this file in the application folder.
- B. Create a web.config file that contains the appropriate configuration code.
Place this file in the same folder as your service contract code.
- C. Create a service file that has the .svc extension containing the @service directive information for the service.
Move this file to the application folder.
- D. Create a service file that has the .svc extension containing the @servicehost directive information for the service.
Move this file to the application folder.
- E. Create a vti_bin sub-folder within the application folder for your code files.
Place the code file that defines and implements the service contract in this folder.
- F. Create an App_Code sub-folder within the application folder for your code files.
Place the code file that defines and implements the service contract in this folder.

Answer: ADF

QUESTION 4

You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5. The service will be hosted on a Web server. You add the following code fragment to the .svc file.

```
<% @ServiceHost Factory="ExamServiceFactory" Service="ExamService" %>
```

You need to create the instances of the services by using the custom ExamServiceFactory class. Which code segment should you use?

- A.

```
public class ExamServiceFactory : ServiceHost
{
    protected override void ApplyConfiguration()
    {
        // Implementation code comes here.
    }
}
```
- B.

```
public class ExamServiceFactory : ServiceHostBase
{
    protected override void ApplyConfiguration()
    {
        // Implementation code comes here.
    }
}
```
- C.

```
public class ExamServiceFactory : ServiceHostFactory
{
    protected override ServiceHost CreateServiceHost(Type serviceType, Uri[] baseAddresses)
    {
        // Implementation code comes here.
    }
}
```

```
D. public class ExamServiceFactory : ServiceHost
{
    public ExamServiceFactory (Type serviceType, params Uri[] baseAddresses) : base(serviceType,
baseAddresses)
    {
        // Implementation code comes here.
    }
}
```

Answer: C

QUESTION 5

You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5. You need to expose two different service endpoints that have the same address. Which configuration setting should you use?

- A. `<service name="ExamService">`
`<endpoint address="http://localhost:8080/service" binding="wsHttpBinding" contract="ISimpleExam"/>`
`<endpoint address="http://localhost:8080/service" binding="wsHttpBinding" contract="IComplexExam"/>`
`</service>`
- B. `<service name="ExamService">`
`<endpoint address="http://localhost:8080/service" binding="wsHttpBinding" contract="ISimpleExam"/>`
`<endpoint address="http://localhost:8080/service" binding="wsDualHttpBinding" contract="IComplexExam"/>`
`>`
`</service>`
- C. `<service name="ExamService">`
`<host>`
`<baseAddresses>`
`<add baseAddress="http://localhost:8080/service"/>`
`</baseAddresses>`
`</host>`
`<endpoint binding="wsHttpBinding" contract="ISimpleExam"/>`
`<endpoint binding="basicHttpBinding" contract="IComplexExam"/>`
`</service>`
- D. `<service name="ExamService">`
`<host>`
`<baseAddresses>`
`<add baseAddress="http://www.certmagic.com/localhost:8080"/>`
`</baseAddresses>`
`</host>`
`<endpoint address="service" binding="wsHttpBinding" contract="ISimpleExam"/>`
`<endpoint address="service" binding="basicHttpBinding" contract="IComplexExam"/>`
`</service>`

Answer: A

QUESTION 6

You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5. You need to host the service in a medium trust environment on a Web server. Which two bindings should you use? (Each correct answer presents a complete solution. Choose two.)

- A. NetMsmqBinding
- B. BasicHttpBinding
- C. WSDualHttpBinding

- D. NetTcpBinding
- E. WebHttpBinding

Answer: BE

QUESTION 7

You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5. You need to programmatically add the following endpoint definition to the service. `http://localhost:8000/ExamService/service`
Which code segment should you use?

- A.

```
String baseAddress="http://localhost:8000/ExamService";
BasicHttpBinding binding1=new BasicHttpBinding();
using(ServiceHost host=new ServiceHost(typeof(ExamService)))
{
    host.AddServiceEndpoint(typeof(IExam),binding1,baseAddress);
}
```
- B.

```
String baseAddress="http://localhost:8000/ExamService/service";
BasicHttpBinding binding1=new BasicHttpBinding();
using(ServiceHost host=new ServiceHost(typeof(ExamService)))
{
    host.AddServiceEndpoint(typeof(IExam),binding1,baseAddress);
}
```
- C.

```
String baseAddress="http://localhost:8000/ExamService";
WsHttpBinding binding1=new WsHttpBinding();
using(ServiceHost host=new ServiceHost(typeof(ExamService)))
{
    host.AddServiceEndpoint(typeof(IExam),binding1,baseAddress);
}
```
- D.

```
String baseAddress="net.tcp://localhost:8000/ExamService/service";
NetTcpBinding binding1=new NetTcpBinding();
using(ServiceHost host=new ServiceHost(typeof(ExamService)))
{
    host.AddServiceEndpoint(typeof(IExam),binding1,baseAddress);
}
```

Answer: B

QUESTION 8

You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5. You write the following code fragment in the service configuration file. (Line numbers are included for reference only.)

```
01 <system.serviceModel>
02 ...
03 <behaviors>
04 <serviceBehaviors>
05 <behavior name="CalculatorServiceBehavior">
06 <CustomServiceBehavior/>
07 </behavior>
08 </serviceBehaviors>
09 </behaviors>
10 ...
11 </system.serviceModel>
```

You need to register the custom service behavior in the service configuration file.

Which code fragment should you insert at line 10?

- A.

```
<behaviorExtensions>
  <add name="CustomServiceBehavior" type="CustomBehavior.CustomServiceBehaviorSection,
  CustomBehavior, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null" />
</behaviorExtensions>
```
- B.

```
<extensions>
  <add name="CustomServiceBehavior" type="CustomBehavior.CustomServiceBehaviorSection,
  CustomBehavior, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null" />
</extensions>
```
- C.

```
<behaviorExtensions>
  <extensions>
    <add name="CustomServiceBehavior" type="CustomBehavior.CustomServiceBehaviorSection,
  CustomBehavior, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null" />
  </extensions>
</behaviorExtensions>
```
- D.

```
<extensions>
  <behaviorExtensions>
    <add name="CustomServiceBehavior" type="CustomBehavior.CustomServiceBehaviorSection,
  CustomBehavior, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null" />
  </behaviorExtensions>
</extensions>
```

Answer: D

QUESTION 9

You are creating an application in Windows Communication Foundation (WCF) by using Microsoft.NET Framework 3.5.

You need to ensure that the client application communicates with the service by using a duplex contract. Which five actions should you perform?

(To answer, move the five appropriate actions from the list of actions to the answer area, and arrange them in the correct order.)

Actions

Create an interface in the WCF service for the serv

Create two interfaces in the WCF service—one for interface and another for the service itself.

Run the WCF service and use the Svcutil.exe applic generate proxies for the client.

Implement the **callback** interface on the **callback**

Create an instance of the callback class and pass it constructor parameter to the **InstanceContext** ob **InstanceContext** object as a constructor paramet client proxy.

Create an instance of the client and use the default configuration as the parameter.

Implement the **service** interface on the **service cl**

ctions

reate an interface in the WCF service for the service i

reate two interfaces in the WCF service—one for a ca
interface and another for the service itself.

un the WCF service and use the Svcutil.exe application
enerate proxies for the client.

implement the **callback** interface on the **callback** class

reate an instance of the callback class and pass it as
onstructor parameter to the **InstanceContext** object
InstanceContext object as a constructor parameter to
lient proxy.

reate an instance of the client and use the default en
onfiguration as the parameter.

implement the **service** interface on the **service** class.

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

Answer:

QUESTION 10

You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5. The service will be hosted in a Console application. You need to configure the service by using a configuration file other than the default app.config file.

Which code segment should you use?

- A.

```
class MyServiceHost : ServiceHost
{
    public MyServiceHost(Type serviceType, params Uri[] baseAddresses) : base(serviceType,
baseAddresses) { }
    protected override void InitializeRuntime() { //Load configuration here }
}
```
- B.

```
class MyServiceHost : ServiceHost
{
    public MyServiceHost(Type serviceType, params Uri[] baseAddresses) : base(serviceType,
baseAddresses) { }
    protected override void ApplyConfiguration() { //Load configuration here }
}
```
- C.

```
class MyServiceHost : ServiceHost
{
    public MyServiceHost(Type serviceType, params Uri[] baseAddresses) : base(serviceType,
baseAddresses) { }
    protected new void InitializeDescription(Type serviceType, UriSchemeKeyedCollection baseAddresses) { //
Load configuration here. }
}
```
- D.

```
class MyServiceHost : ServiceHost
{
    public MyServiceHost(Type serviceType, params Uri[] baseAddresses) : base(serviceType,
baseAddresses) { }
    protected new void AddBaseAddress(Uri baseAddress) { //Load configuration here. }
}
```

Answer: B