

## [Pass Ensure VCE Dumps Training PassLeader New 70-513 341q VCE and PDF Dumps To Pass Exam (41-60)

Being tormented to get well prepared for your 70-513 exam? Don't worry! PassLeader now offer the first-hand 70-513 exam dumps, you will pass 70-513 exam for your first try with PassLeader's latest real 341q 70-513 exam questions. We offer you the newest 70-513 exam study guide with VCE test engine or PDF format braindumps, you can get the basic knowledge and all details about 70-513 exam. Do not hesitate to try our high quality 341q 70-513 practice tests! keywords: 70-513 exam,341q 70-513 exam dumps,341q 70-513 exam questions,70-513 pdf dumps,70-513 practice test,70-513 vce dumps,70-513 study guide,70-513 braindumps,TS: Windows Communication Foundation Development with Microsoft .NET Framework 4 Exam



QUESTION 41 You create a Windows Communication Foundation (WCF) service. You need to ensure that specific users can enable message logging for the service at runtime. In the configuration file for the service, what should you do?

- A. Enable debug compilation.
- B. Enable a shared XMLTraceListener object.
- C. Enable the Windows Management Instrumentation (WMI) provider.
- D. Enable message logging.

Answer: A QUESTION 42 You are developing a Windows Communication Foundation (WCF) service that must be discoverable. You need to ensure that the ServiceHost instance supports multiple discovery versions. What should you do?

- A. Specify a unique DiscoveryVersion parameter for each endpoint constructor. Use the same value for the Address property of each endpoint.
- B. Use the endpoint constructor without the DiscoveryVersion parameter. Use a unique value for the Address property of each endpoint.
- C. Specify a unique DiscoveryVersion parameter for each endpoint constructor. Use a unique value for the Address property of each endpoint.
- D. Use the endpoint constructor without the DiscoveryVersion parameter. Use the same value for the Address property of each endpoint.

Answer: C Explanation: We should specify version discovery for each endpoint constructor and use unique addresses for them. <http://msdn.microsoft.com/en-us/library/dd456799.aspx> QUESTION 43 A service implements the following contract. (Line numbers are included for reference only.)

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

The service is implemented as follows.

```
20 class ContosoService : IContosoService
21 {
22     public void OperationOne(string value) { ... }
23
24     public void OperationTwo(string value) { ... }
25 }
```

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. Answer: B QUESTION 44 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 Answer: D QUESTION 45 A Windows Communication Foundation (WCF) service is self-hosted in a console application. The service implements the IDataAccess contract, which is defined in the MyApplication namespace. The service is implemented in a class named DataAccessService which implements the IDataAccess interface and also is defined in the MyApplication namespace. The hosting code is as follows. (Line numbers are included for reference only.) 01 static void Main(string[] args) 02 { 03 ServiceHost host; 04 ... 05 host.Open(); 06 Console.ReadLine(); 07 host.Close(); 08 } You need to create a ServiceHost instance and assign it to the host variable. You also need to instantiate the service host. Which line of code should you insert at line 04? A. host = new ServiceHost("MyApplication.DataAccessService"); B. host = new ServiceHost("MyApplication.DataAccess"); C. host = new ServiceHost(typeof(IDataAccess)); D. host = new ServiceHost(typeof(DataAccessService)); Answer: D QUESTION 46 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> Answer: B QUESTION 47 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); Answer: A Explanation: Claim Set can has only one Rights. Identity claim. It contains user identity information. All additional data, like DNS, stored in claims with Possess Property rights. QUESTION 48 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. Answer: B QUESTION 49 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> Answer: A  
QUESTION 50 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 Answer: D

Why Not Try PassLeader New Premium 70-513 Exam Dumps?

Product	Status	Q&As	Price
Pass4sure	Banned By Microsoft Not Available		
PassLeader	Leader of IT Certifications BONUS !!! Free VCE Player	341 Q&As	Price: \$99.99
TEST KING		153 Q&As	Price: \$124.99

Coupon Code -- CELEB

<http://www.passleader.com/70-513.html> QUESTION 51 Your company has an existing Windows Communication Foundation (WCF) service. The following code segment is part of the service. (Line numbers are included for reference only.)

```
01 ServiceHost h  
02 www.p  
host.Open();
```

- You need to ensure that AJAX client applications can access the service. Which code segment should you insert at line 02?
- A. Net: Ser: ty: ep. I
  - B. Web: Ser: ty: ep. I
  - C. Net: Ser: ty: ep. I
  - D. Bas: Ser: ty: ep. I

A. Option A B. Option B C. Option C D. Option D Answer: B  
QUESTION 52 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. D. Create a class with the same name as the previously generated proxy and add the new asynchronous methods as partial methods. Answer: B  
QUESTION 53 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) { return null; } C.
public void AfterCall(string operationName, object[] outputs, object returnValue, object correlationState) { String
accountNumber = GetAccountNumber(outputs); var validator = new AccountNumberValidator(); if(
!validator.Validate(accountNumber)) { return value = 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(); } } Answer: D
```

QUESTION 54 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"); Answer: D
```

QUESTION 55 A Windows Communication Foundation (WCF) client configuration file contains the following XML segment in the system.serviceModel element.

```
<client>
<endpoint address="net.tcp://server/ContosoService" binding="netTcpBinding" contract="Contoso.IContosoService"
name="netTcp"/>
<endpoint address="net.pipe://localhost/ContosoService" binding="netNamedPipeBinding"
contract="Contoso.IContosoService" name="netPipe" />
</client>
```

You need to create a channel factory that can send messages to the endpoint listening at net.pipe://localhost/ContosoService. Which code segment should you use? A. ChannelFactory<Contoso.IContoso> factory = new ChannelFactory<Contoso.IContoso>("Contoso.IContoso");

```
B. ChannelFactory<Contoso.IContoso> factory = new
ChannelFactory<Contoso.IContoso>("netNamedPipeBinding"); C. ChannelFactory<Contoso.IContoso>
factory = new ChannelFactory<Contoso.IContoso>("netPipe"); D. ChannelFactory<Contoso.IContoso>
factory = new ChannelFactory<Contoso.IContoso>("net.pipe//localhost/ContosoService"); Answer: C
```

QUESTION 56 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 Answer: CD
```

QUESTION 57 You are developing a Windows Communication Foundation (WCF) service that does not operate on a duplex channel. You find that operations do not start until all previous operations have finished. The service hosting code contains the following lines.

```
var service = new WarehouseService(); var host = new ServiceHost(service); You need to ensure that new operations do not wait
for previous operations to finish. Which attribute should you use to decorate the service? A. [ServiceBehavior(InstanceContextMode=InstanceContextMode.Single, ConcurrencyMode=ConcurrencyMode.Multiple)]
```

```
B. [CallbackBehavior(ConcurrencyMode=ConcurrencyMode.Multiple)] C. [ServiceBehavior(InstanceContextMode=InstanceContextMode.Single,
ConcurrencyMode=ConcurrencyMode.Single)]
D. [ServiceBehavior(InstanceContextMode=InstanceContextMode.Single,
ConcurrencyMode=ConcurrencyMode.Reentrant)] Answer: A
```

Explanation: Only Concurrency Mode. Multiple gives as singleton service with support of multiple requests

QUESTION 58 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 IogKnownPii 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 IogKnownPii 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> Answer: D QUESTION 59 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 4.0.0.0 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" /> Answer: A QUESTION 60 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
07     [OperationContract]
08     void PutMessage(string message);
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 }

```

[www.passleader.com](http://www.passleader.com)

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

```

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

```

[www.passleader.com](http://www.passleader.com)

You need to ensure that all clients calling GetMessage will retrieve the updated string if the message is updated by any client 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; } Answer: A



<http://www.passleader.com/70-513.html>