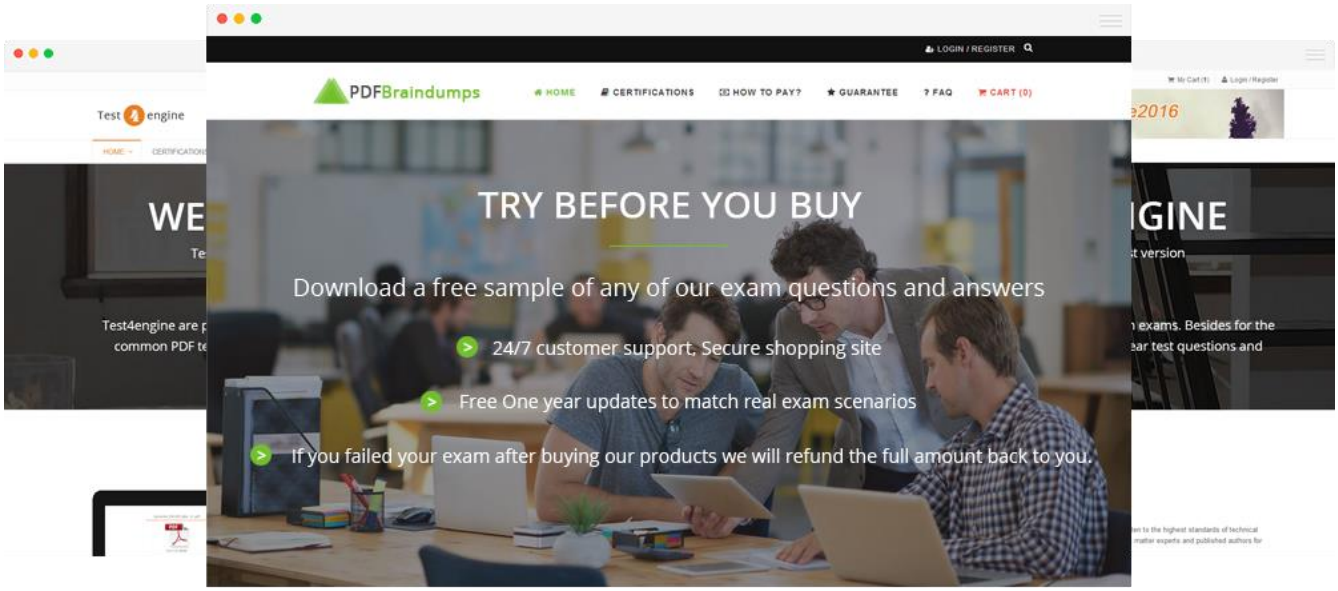
PDFBraindumps



Latest Pdf Braindumps

- LRP-614
- C-SRM-72
- BCABA
- ACMP-6.3
- JN0-740
- 1z0-100
- 250-405
- FM0-308
- DS-200
- C_SRM_72
- SDM_2002001040
- ACMP_6.3
- ST0-250
- 1z0-062
- H12-221
- A4120-784
- M2180-716

Top Certifications

- ISEB Certification
- HCNP
- OCE
- IFPUG Certifications
- NVIDIA Certifications
- dotMobi Certification
- Network+
- SCMA
- IBM Certified Integrat
- MCSD
- CCDH
- NCLP
- IBM Certified Advanc
- XMLMaster Certificat
- eserver Certified Spe
- CS5
- SAP-Certifications
- CHA
- Network Appliance N

Top Vendors

- ISEB
- Fortinet
- ASTQB
- Ericsson
- Aruba
- Liferay
- Data Center Universit
- Novell
- HRCI
- Huawei
- CIW
- RSA
- Patchlink
- MYSQL
- International Consorti
- ISM
- Acme-Packet
- CheckPoint

http://www.pdfbraindumps.com

Latest pdf braindumps provider, high pass rate

Exam : **111-056**

Title : TS:MS.NET Framework 2.0 - Distributed Application Developm

Vendors : Microsoft

Version : DEMO

NO.1 You create a .NET Framework remoting application that provides stock information to customers. The server component raises an event on the client computer when certain conditions are met. You need to ensure the server raises exactly one event for each client application that is registered for the event. What should you do?
A. Configure the server class as a Singleton Server Activated Object (SAO) and check for duplicate client delegate methods before raising the event.
B. Configure the server class as a Client Activated Object (CAO) and override the CreateObjRef method to check for duplicate client delegate methods before raising the event.
C. Configure the server class as a SingleCall Server Activated Object (SAO) and check for duplicate client delegate methods before raising the event.
D. Configure the server class as a Client Activated Object (CAO) and check for duplicate client delegate methods before raising the event.
Answer: A

NO.2 You are converting an application to use .NET Framework remoting. The server portion of the application monitors stock prices and contains a class named StockPriceServer, which is a Server Activated Object (SAO). The client computer interacts with the server using a common assembly. When the server attempts to raise an event on the client computer, the server throws the following exception.System.IO.FileNotFoundException.You discover that the event delegate is not being called on the client computer. You need to ensure that the server application can raise the event on the client computer. What should you do?
A. Add the Serializable attribute to the StockPriceServer class and change the event to use one of the standard common language runtime (CLR) delegates.
B. In the common assembly, add an interface that contains the event and a method to raise the event. Implement that interface in the StockPriceServer class and use the interface's event to register the delegate message on the client computer.
C. Add the event delegate to the common assembly. Implement the Add delegate and the Remove delegate methods of the event in the StockPriceServer class to reference the delegate method in the client application.
D. Raise the event using the BeginInvoke method and pass a reference to the client computer.
Answer: B

NO.3 A class library named MathLib contains the following code.public class MathClass : MarshalByRefObject { public decimal DoHugeCalculation(int iterations) { decimal result; //Some very lengthy calculations ... return result; }}The MathLib class is hosted in a .NET Framework remoting server application. A Windows application project running on a client computer contains the following class.public class MathClient { public void ProcessHugeCalculation(int iterations)
{ MathClass cm = new MathClass(); decimal decRes = cm.DoHugeCalculation(iterations); //process the result ... }}The MathClient class must call the MathClass class asynchronously by using remoting. A callback must be implemented to meet this requirement. You need to complete the implementation of the MathClient class. What should you do?
A. Modify the MathClient class as follows:public class MathClient {public delegate void DoHugeCalculationDelegate(decimal result);public event DoHugeCalculationDelegate

DoHugeCalculationResult;public void DoHugeCalculationHandler(decimal result) {DoHugeCalculationResult(result);} public void ProcessHugeCalculation(int iterations) { //Hook up event handler here... ... }}

B. Apply the Serializable attribute to the MathClient class.

C. Modify the MathClient class as follows:public class MathClient { private delegate decimal DoHugeCalculationDelegate(int iterations); private void DoHugeCalculationCallBack(IAsyncResult res) { AsyncResult aRes = (AsyncResult)res; decimal decRes = ((DoHugeCalculationDelegate)aRes. AsyncDelegate).EndInvoke(res); //process the result ... } public void ProcessHugeCalculation(int iterations) { MathClass cm = new MathClass(); DoHugeCalculationDelegate del = new DoHugeCalculationDelegate(cm.DoHugeCalculation); del.BeginInvoke(iterations, new AsyncCallback(DoHugeCalculationCallBack), null); }}

D. Apply the OneWay attribute to all methods in the MathClass class.

Answer: C