

TestkingIT

Testking IT

> Contact Us Login / Register Search...

- HOME
- ALL VENDORS
- ★ GUARANTEE
- ? FAQ
- TESTIMONIALS
- CART (0)



Try **Desktop Test Engine** before you buy

We're not the only ones **happy** about TestKingsIT Practice Material ...

48236+ customers in 100+ countries use TestKingsIT Test Engine. Meet our customers.

- iMessenger
- VOREED
- GetCustom
- JET ORANGE
- iCompany
- Paradoxx



<http://www.testkingit.com/>

Latest practice material - Exam Cram - TestKingIT

Exam : 70-306

Title : Developing and Implementing
Windows-based Applications with
Microsoft Visual Basic .NET

Vendors : Microsoft

Version : DEMO

NO.1 You use Visual Studio .NET to create a component that will be shared by two client applications.

Eventually, you plan to deploy new versions of this shared component. However, not all of the new

versions will be compatible with both client applications. When you deploy the shared component and the

client applications, you must ensure that you can upgrade the shared component for a single client

application. You must also minimize the need for configuration changes when you deploy new versions of

the component. What are two possible ways to achieve your goal? (Each correct Answer presents a complete solution. Choose two.)

A. Deploy each client application to its own folder. Deploy the shared component to its own folder.

Register the shared component by using RegSvr32 with the /s option.

B. Deploy each client application to its own folder. Deploy a separate copy of the shared component to

each client application folder. When you deploy a new version of the component, replace the older

version only if the new version remains compatible with the client application in the same folder.

C. Compile the client applications with references to the shared component. Deploy both client

applications and the shared component to a single folder. When you deploy a new version of the

component, increment its version number.

D. Create a strong name for the shared component and specify a version number. Compile each client

application and bind it to the shared component. Deploy the shared component to the global assembly

cache on the client computer. Deploy each client application to its own folder. When you deploy a new

version of the component, increment its version number.

Answer: BD

NO.2 You use Visual Studio .NET to create an assembly that will be used by other applications, including a

standard COM client application. You must deploy your assembly and the COM application to

a client

computer. You must ensure that the COM application can instantiate components within the assembly as

COM components. What should you do?

A. Create a strong name for the assembly by using the Strong Name tool (Sn.exe).

B. Generate a registry file for the assembly by using the Assembly Registration tool (Regasm.exe).

Register the file on the client computer.

C. Generate a type library for the assembly by using the Type Library Importer (Tlbimp.exe).

Register the

file on the client computer.

D. Deploy the assembly to the global assembly cache on the client computer. Add a reference to the

assembly in the COM client application.

Answer: B

NO.3 Your development team uses Visual Studio .NET to create an accounting application, which contains a

class named SupplierAccounts. This class instantiates several classes from a COM component that was

created by using Visual Basic

6.0. Each COM component class includes a custom method named ShutDownObject that must be called

before terminating references to the class. Software testers report that the COM component appears to

remain in memory after the application terminates. You must ensure that the

ShutDownObject method of

each COM component class is called before SupplierAccounts is terminated. What should you do?

A. Add code to the Terminate event of SupplierAccounts to call the ShutDownObject method of each

COM component

class.

B. Find each location in your code where a reference to SupplierAccounts is set to Nothing or goes out of

scope. Add code after each instance to manually invoke the Visual Studio .NET garbage collector.

C. Add the procedure Protected Overrides Finalize() to SupplierAccounts. Add code to the

procedure to

call the `ShutDownObject` method of each COM component class.

D. Add the procedure `Private Sub Finally()` to `SupplierAccounts`. Add code to the procedure to call the

`ShutDownObject` method of each COM component class.

Answer: C

NO.4 You use Visual Studio .NET to create an application that interacts with a Microsoft SQL Server

database. You create a SQL Server stored procedure named `CustOrderDetails` and save it in the

database. Other developers on your team frequently debug other stored procedures. You need to verify

that your stored procedure is performing correctly. You need to step through

`CustOrderDetails` inside the

Visual Studio .NET debugger. What should you do?

A. Run `CustOrderDetails` by using the Visual Studio .NET Server Explorer.

B. Step into `CustOrderDetails` by using the Visual Studio .NET Server Explorer.

C. From the Command window, use `Ctrl+E` to run `CustOrderDetails`.

D. Move `CustOrderDetails` from the Visual Studio .NET Server Explorer window to a Windows Form. Run

the application in Debug mode and step through `CustOrderDetails`.

Answer: B

NO.5 You develop a Windows-based application named `CustOrders`. You implement the `Trace` object within

your application code. You will use this object to record application information, such as errors and

performance data, in a log file. You must have the ability to enable and disable `Trace` logging.

This

functionality must involve the minimum amount of administrative effort. What should you do?

A. Create a Boolean constant in your application named `TraceLogging` and set it to `False`.

Each time your

code uses `Trace` logging, use a `If Then` statement to evaluate your `#TraceLogging` constant.

B. On each computer that will host your application, create an environment variable named `CustOrders.Trace`. Set the environment variable to `True` when you want to enable `Trace` logging. Set it to

`False` when you want to disable `Trace` logging.

C. On each computer that will host your application, edit the shortcut used to start your application.

Add/d:TRACE=True to the Target property.

D. Use the TraceSwitch class within your code. Each time your code uses Trace logging, consult the

TraceSwitch level to verify whether to log information. Change the TraceSwitch level by editing your

application's .config file.

Answer: D

NO.6 You use Visual Studio .NET to create a Windows-based data management application named MyApp.

You implement the following code segment: Dim oSwitch As New _TraceSwitch("MySwitch", "My

TraceSwitch") Dim oWriter As New _IO.StreamWriter(IO.File.Open("c:\MyApp.txt", _

IO.FileMode.Append)) Dim oListener As New _TextWriterTraceListener(oWriter)

Trace.Listeners.Add(oListener) Try CustomerUpdate() Catch oEx As Exception

Trace.WriteLineIf(oSwitch.TraceError, _ "Error: " & oEx.Message) Finally T

Trace.Listeners.Clear()

oWriter.Close() oWriter = Nothing

End Try You compile a debug version of the application and deploy it to a user's computer.

The user

reports errors, which are generated within the CustomerUpdate procedure. You decide to enable logging

of the error messages generated by CustomerUpdate. You want to use the minimum amount of

administrative effort. What should you do?

A. Start the application with the following command line: /TRACE MySwitch 1.

B. Start the application with the following command line: /d:TRACE=True.

C. Start the application with the following command line: /XML <switch="MySwitch" value="1">

D. Create an environment variable on the user's computer. Name the variable MySwitch and assign it a

value of 1.

E. Edit your application's .config file to set the value of MySwitch to 1.

Answer: E

NO.7 You develop a Windows-based application by using Visual Studio .NET and Microsoft

SQL Server.

The application will perform numerous Assert, Deny, and PermitOnly security operations while it is

executing. You must ensure that the application is optimized for fast run-time execution.

What should you

do?

- A. Perform declarative security checks.
- B. Perform imperative security checks.
- C. Perform all security checks by using SQL Server security.
- D. Implement a custom security class that retrieves security information from the SQL Server database.

Answer: A

This document was created with Win2PDF available at <http://www.win2pdf.com>.

The unregistered version of Win2PDF is for evaluation or non-commercial use only.

This page will not be added after purchasing Win2PDF.

NO.8 Your project team uses Visual Studio .NET to create an accounting application. Each team member

uses the Writemethod of both the Debug class and the Trace class to record information about application

execution in the Windows2000 event log.

You are performing integration testing for the application. You need to ensure that only one entry is added

to the eventlog each time a call is made to the Write method of either the Debug class or the Trace

class.What are two possible code segments for you to use? (Each correct

Answer presents a complete solution. Choose two.)

- A. Dim myTraceListener As New _ EventLogTraceListener("myEventLogSource")
Trace.Listeners.Add(myTraceListener)
- B. Dim myDebugListener As New _ EventLogTraceListener("myEventLogSource")
Debug.Listeners.Add(myDebugListener)
- C. Dim myTraceListener As New _ EventLogTraceListener("myEventLogSource")
Debug.Listeners.Add(myTraceListener) Trace.Listeners.Add(myTraceListener)
- D. Dim myDebugListener As New _ EventLogTraceListener("myEventLogSource") Dim
myTraceListener
As New _EventLogTraceListener("myEventLogSource")
Debug.Listeners.Add(myDebugListener)
Trace.Listeners.Add(myTraceListener)

Answer: AB

NO.9 Another developer in your company uses Visual Studio .NET to create a component named MyComponent. You deploy MyComponent to a server. When you execute MyComponent, you receive the following error message: "System.Security.Policy.PolicyException: Failed to acquire required permissions." As quickly as possible, you need to discover which permissions are required by MyComponent. What should you do?

- A. Request the source code from the developer who created MyComponent. Examine the source code to find the required permissions.
- B. Run the Microsoft CLR Debugger (DbgCLR.exe) on the server to view the permissions requested by the application at run time.
- C. Run the Runtime Debugger (Cordbg.exe) on the server to view the permissions requested by the application at run time.
- D. Run the Permissions View tool (Permview.exe) on the server to view the permissions required by MyComponent.
- E. Run the MSIL Disassembler (Ildasm.exe) on the server to view permission requests by MyComponent that were denied.

Answer: D

NO.10 You are maintaining a Visual Studio .NET sales application that was created by another developer. The application functions as expected for several months. Then users report that it sometimes calculates tax amounts incorrectly. An examination of the source code leads you to suspect that the errors are caused by a function named CalculateSalesTax. To test your hypothesis, you place a breakpoint on the following line of code: `decTax = CalculateSalesTax(decRate, decSaleAmount)` However, when you run the application to create a test invoice, the breakpoint is not invoked. How should you correct this

problem?

- A. Select Enable All Breakpoints from the Debug menu.
- B. Select Configuration Manager from the Build menu. Set the Active Solution Configuration option to Debug . Set the Configuration property of the project to Debug .
- C. Select Options from the Tools menu and then select the General object from the Debugging folder. Choose the option In break mode, only stop execution of the current process .
- D. Select Exceptions from the Debug menu. Under the heading If the exception is not handled , select Break into the Debugger .

Answer: B