

GlobalTcert

GLOBALTCERT.COM
Your gateway to success



Demo
STUDY GUIDE

© Copy Right 1998-2005 GlobalTcert LLC. All Rights Reserved.

QUESTION 1

You develop an application named gtApp. This application needs to run on the same computer as a Windows service named gtService.

You want to ensure that gtService starts from gtApp if gtService is not already running.

Which code segment should you use?

- A.

```
ServiceController gtServiceController = new
ServiceController("gtService");
if (gtServiceController.Status == ServiceControllerStatus.Stopped)
{ gtServiceController.Start(); }
```
- B.

```
ServiceController gtServiceController = new
ServiceController("gtService"); gtServiceController.Start();
```
- C.

```
String[] gtArgs = new string [1];
gtArgs[0] = "gtService";
ServiceController gtServiceController = new ServiceController();
if (gtServiceController.Status == ServiceControllerStatus.Stopped)
{ gtServiceController.Start(gtArgs); }
```
- D.

```
String[] gtArgs = new string[1];
gtArgs[0] = "gtService";
ServiceController gtServiceController = new ServiceController();
myServiceController.Start(gtArgs);
```

Answer: A

QUESTION 2

Your Microsoft SQL Server database contains a table named GlobaliT Orders. GlobaliT Orders is used to store new purchase orders as they are entered into an order-entry application. To keep up with customer demand, the order fulfillment department wants to know at 15-minute intervals when new orders are entered.

You need to develop an application that reads GlobaliT Orders every 15 minutes and sends all new orders to the order fulfillment department. The application will run on computer that is used by several users who continuously log on and log off from the network to perform miscellaneous tasks.

Which type of .NET application should you use?

- A. Windows Form
- B. Windows service
- C. XML Web service
- D. .NET Remoting object

Answer: B

Explanation: A Windows service would still be running even though users logs on and off.

Incorrect Answers

A: A Windows Form would be closed when a user logs off.

C: An XML Web service is not guaranteed to keep running if a user logs off.

D: You can use .NET Remoting to enable different applications to communicate with one another. However, a remoting object would be destroyed when a user logs off the system.

QUESTION 3

You create a serviced component named GlobaliT Item that implements an interface named I GlobaliT Item. You want to ensure that calls to the serviced component through I GlobaliT Item are queued. What should you do?

A. To GlobaliT Item, add the following attribute:

[InterfaceQueuing(true, Interface="I GlobaliT Item")] B. To GlobaliT Item, add the following attribute:

[Transaction(TransactionOption.Disabled, Isolation=TransactionIsolationLevel.Serializable)]

C. To the GlobaliT Item assembly, add the following attribute:

[assembly: ApplicationQueuing(Enabled=true, QueueListenerEnabled=true)]

D. In the GlobaliT Item implementation, override the Activate method from the ServicedComponent class. In the Activate method, add the following code segment:

```
Queue q = new Queue(); q.Enqueue(this);
```

Answer: C

Explanation: In addition to enabling queued component support at the application level, you must mark your interfaces as capable of receiving queued calls. You do that by using setting the QueueListenerEnabled attribute to True.

Note: The COM+ Queued Components (QC) service provides an easy way to invoke and execute components asynchronously. Processing can occur without regard to the availability or accessibility of either the sender or receiver.

Reference: .NET Framework Developer's Guide, Queued Components [C#]

Incorrect Answers

A: The signature for the InterfaceQueuing attribute as shown in Answer A is wrong.

B: Transactions are not helpful for interface queuing configuration.

D: Creating a new queue in the Active method is not correct in this scenario.

QUESTION 4

You are creating a serviced component named ItemGTInventory. An online catalog application will use ItemGTInventory to display the availability of products in inventory.

Additional serviced components written by other developers will continuously update the inventory data as orders are placed.

The ItemGTInventory class includes the following code segment:

```
<Transaction(TransactionOption.Required)> _
```

```
Public Class ItemGTInventory
```

```
Inherits ServicedComponent
```

```
' Method code goes here.
```

```
End Class
```

ItemInventory is configured to require transactions. You want ItemInventory to respond to requests

as quickly as possible, even if that means displaying inventory values that are not up to date with the most recent orders.

What should you do?

A. To the ItemGTInventory class, add the following attribute:

<ObjectPooling(True)>

B. To all methods of the ItemGTInventory class, add the following attribute:

<AutoComplete(False)>

C. Modify the Transaction attribute of the ItemGTInventory class to be the following attribute:

<Transaction(TransactionOption.Required, Timeout:=1)>

D. Modify the Transaction attribute of the ItemGTInventory class to be the following attribute:

<Transaction(TransactionOption.Required, _
IsolationLevel:=
TransactionIsolationLevel.ReadUncommitted)>

Answer: D

Explanation: The ReadUncommitted transaction isolation level makes a dirty read is possible, meaning that no shared locks are issued and no exclusive locks are honored. This will reduce the number of locks, compared to the default transaction isolation level of readcommitted. Reduced number of locks will increase the performance.

Incorrect Answers

A: Object pooling is a COM+ service that enables you to reduce the overhead of creating each object from scratch. However, object pooling is not much use in transactions.

B: Autocomplete does not apply here.

C: Timeout configuration would not address performance issue.

Reference: .NET Framework Class Library, IsolationLevel Enumeration

QUESTION 5

You are creating a serviced component named UserManager. UserManager adds user accounts to multiple transactional data sources.

The UserManager class includes the following code segment:

```
[Transaction(TransactionOption.Required)]  
[SecurityRole("Admin")]  
public class UserManager : ServicedComponent {  
    public void AddUser(string GlobaliT name, string GlobaliT password) {  
        // Code to add the user to data sources goes here.  
    }  
}
```

You must ensure that the AddUser method reliably saves the new user to either all data sources or no data sources.

What should you do?

A. To AddUser, add the following attribute:

[AutoComplete()]

B. To UserManager, add the following attribute: