

GlobalTcert.com

GLOBALTCERT.COM
Your gateway to success



Demo
STUDY GUIDE

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

QUESTION 1

You create a Windows service named GTService that requires a table named Orders in a Microsoft SQL Server database. You want GTService to check every 30 seconds for new rows in Orders.

You create the following method in myService:

```
Private Sub ProcessOrders(ByVal source As Object, ByVal  
eventArguments As Timers.ElapsedEventArgs)
```

```
'Code to process orders goes here.
```

```
End Sub
```

You need to add additional code to GTService to invoke the ProcessOrders method.

What should you do?

A. To the OnStart method, add the following code segment:

```
Dim GTTimer As New Timers.Timer()
```

```
AddHandler GTTimer.Elapsed, AddressOf ProcessOrders
```

```
GTTimer.Interval = 30000
```

```
GTTimer.Enabled = True
```

B. To the OnCustomCommand method, add the following code segment:

```
Dim GTTimer As New Timers.Timer()
```

```
AddHandler GTTimer.Elapsed, AddressOf ProcessOrders
```

```
GTTimer.Interval = 30000
```

```
GTTimer.Enabled = True
```

C. To the OnStart method, add the following code segment:

```
Dim GTTimer As New Timers.Timer()
```

```
AddHandler GTTimer.Elapsed, AddressOf ProcessOrders
```

```
GTTimer.Interval = 30000
```

```
GTTimer.AutoReset = True
```

D. To the OnCustomCommand method, add the following code segment:

```
Dim GTTimer As New Timers.Timer()
```

```
AddHandler GTTimer.Elapsed, AddressOf ProcessOrders
```

```
GTTimer.Interval = 30000
```

```
GTTimer.AutoReset = True
```

Answer: A

Explanation:

Use OnStart to specify the processing that occurs when the service receives a Start command. OnStart is the method in which you specify the behavior of the service. OnStart can take arguments as a way to pass data, but this usage is rare.

After configuration of the timer we must start it by raising the Enabled event.

Reference:

.NET Framework Class Library, ServiceBase.OnStart Method [Visual Basic]

.NET Framework Class Library, Timer Members

.NET Framework Class Library, ServiceBase.OnCustomCommand Method [Visual Basic]

Incorrect Answers

B, D: OnCustomCommand executes when the Service Control Manager (SCM) passes a custom command to the service. The OnCustomCommand is not a good place to place define and configure a timer.

C: The `Timer.AutoReset` property indicates whether the `Timer` should raise the `Elapsed` event each time the specified `Interval` elapses or only after the first time it elapses. The default value is `True`, so there is no need to set to true in this scenario. Furthermore, the timer will not start until the `Enabled` property is set to true.

QUESTION 2

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.

```
Dim GTServiceController As New _
ServiceController("GTService")
If GTServiceController.Status = _
ServiceControllerStatus.Stopped
Then GTServiceController.Start()
End If
```
- B.

```
Dim GTServiceController As New _
ServiceController("GTService")GTServiceController.Start()
```
- C.

```
Dim GTServiceController As New _
ServiceController() Dim GTArgs(1)As String GT
Args(0)="GTService"
If GTServiceController.Status = ServiceControllerStatus.Stopped
Then GTServiceController.Start(GTArgs)
End If
```
- D.

```
Dim GTServiceController As New _
ServiceController() Dim GTArgs(1) As String CK Args(0)= "GTService"
GTServiceController.Start(GTArgs)
```

Answer: A

Explanation: First we create a new instance of `GTService`. Then we check the current state of it with the `Status` property. Finally we use the `ServiceController` start method to start it if it was stopped..

Note: A `Service Controller` object represents a Windows service and allows you to connect to a running or stopped service, manipulate it, or get information about it.

Reference:

- .NET Framework Class Library, `ServiceController` Class
- .NET Framework Class Library, `ServiceController` Members
- .NET Framework Class Library, `ServiceControllerStatus` Enumeration [Visual Basic]
- .NET Framework Class Library, `ServiceController` Constructor [Visual Basic]
- Visual Basic and Visual C# Concepts, Creating `ServiceController` Component Instances

Incorrect Answers

B: The constructor is incorrect.

C, D: Here the `ServiceController` constructor is used with two parameters. The first parameter correctly references the existing server `GTService`. The second parameter, 'GT', specifies the computer that the

service runs on. CK has no special significance (we don't know the name of the computer). Furthermore, we do not have to specify the computer name if the service runs on the local computer.

QUESTION 3

Your Microsoft SQL Server database contains a table named GlobalitcerT Orders. GlobalitcerT 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 GlobalitcerT 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 4

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

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

The ItemInventory class includes the following code segment:

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

```
Public Class ItemInventory
```

```
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 ItemInventory class, add the following attribute:

```
<ObjectPooling(True)>
```

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