

eLearning

anytime,



anywhere  
education

# EliteCertify

LEADING THE WAY TO SUCCESS

70-310 DEVELOPING XML WEB SERVICES AND SERVER  
COMPONENTSWITH MICROSOFT VISUAL BASIC .NET DEMO



© Copyright 1997-2003, EliteCertify.com, All Rights Reserved.

You have an ASP.NET application named TestKWebApp. This application uses a private assembly named Employee to store and retrieve employee data. Employee is located in the bin directory of TestKWebApp.

You develop a new ASP .NET application named TestKWebApp2 that also needs to use employee. You assign Employee a strong name, set its version to 1.0.0.0, and install it in the global assembly cache. You then create a publisher policy assembly for version 1.0.0.0 and install it in the global assembly cache.

You complete TestKWebApp2 against version 1.0.0.0. You do not recompile TestKWebApp. You then run MyWebApp.

What is the most likely result?

- A. A VersionNotFoundException is Thrown.
- B. Employee is loaded from the bin directory.
- C. Version 1.0.0.0 of Employee is loaded from the global assembly cache.
- D. Version 1.0.0.0 of Employee is loaded by the publisher policy assembly.

**Answer: D**

**Explanation:** Vendors of assemblies can state that applications should use a newer version of an assembly by including a publisher policy file with the upgraded assembly.

**Reference:**

.NET Framework Developer's Guide. Creating a Publisher Policy File  
.NET Framework Developer's Guide, Versioning

**Incorrect Answers**

- A:** A VersionNotFoundException represents the exception that is thrown when attempting to return a version of a DataRow that has been deleted.
- B, C:** The Publisher Policy Assembly will be used.

You create an XML Web service named TestKService. You must ensure that this service meets the following URL authorization requirements.

- Anonymous access must be disabled for TestKService.
- An authenticated user named User1 cannot access TestKService.
- All other authenticated users can access TestKService.

You configure Internet Information Services (IIS) to meet these requirements. You now need to configure the authorization section in the Web.config file to properly authorize the users.

**Which code segment should you use?**

- A. `<allow users="*" />`  
`<deny users="User1" />`
- B. `<allow users="?" />`  
`<deny users="User1" />`
- C. `<deny users="*" />`  
`<deny users="User1" />`  
`<allow users="?" />`
- D. `<deny users="?" />`  
`<deny users="User1" />`  
`<allow users="*" />`

**Answer: D**

**Explanation:** The elements are applied in order one by one. The first matching element decides whether the user should be granted access or not.

**Step 1:** We deny access permissions to anonymous users with `<deny users="?" />`

**Step 2:** We deny access to User1 with `<deny users="User1" />`

**Step 3:** We allow access for all other users with `<allow users="*" />`

**Note:**

Identity	Description
*	Refers to all identities
?	Refers to the anonymous identity

**Reference:** .NET Framework Developer's Guide, ASP.NET Authorization

**Incorrect Answers**

**A:** All users are allowed access. We must deny before we allow.

**B:** All anonymous users are allowed access.

**C:** All users are denied access.

**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
    Inherits ServicedComponent
    Public Sub AddUser (ByVal name As String, _
        ByVal password As String)
        ` Code to add the user to data sources goes here.
    End Sub
End Class

```

**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:  
<JustInTimeActivation (False) >
- C. To the end of AddUser, add the following line of code:  
ContextUtil.EnableCommit ( )
- D. To the end of AddUser, add the following line of code:  
ContextUtil.MyTransactionVote = True

**Answer: C**

**Explanation:** The `TransactionOption.Required` shares a transaction, if one exists, and creates a new transaction, if necessary. We should commit this transaction at the end of `AddUser` with `ContextUtil.EnableCommit ( )` statement.

**Reference:** .NET Framework Class Library, ContextUtil Methods

**Incorrect Answers**

**A:** Autocomplete is out of context here.

**B:** Just-in-time (JIT) activation is a COM+ service that enables you to create an object as a nonactive, context-only object. JIT does not apply here.

**D:** Not useful.