

Building COM+ Applications Using Microsoft .NET Enterprise Services

Course 2557—Five days—Instructor-led

Introduction

This five-day, instructor-led course provides students with the knowledge and skills to effectively build scalable, distributed applications that use Microsoft .NET Enterprise Services and the Microsoft .NET Framework.

Audience

This course is intended for:

- Departmental developers who currently build client/server applications and large-scale solutions for departments inside a business.
- Consultants who work with developers to build client/server applications and large-scale solutions for businesses.
- Web developers who build Web-based applications requiring significant infrastructure support such as transactions, security, and pooling of resources.
- Line-of-business (LOB) application developers.

At Course Completion

After completing this course, students will be able to:

- Describe the history of server-based applications and describe the COM+ runtime architecture.
- Use attributes to configure an assembly as a COM+ application and create components that use just-in-time (JIT) activation and are synchronized.
- Create data access components that use Microsoft ADO.NET to retrieve or update data in a database.
- Create transactional components.
- Help protect an enterprise application by using COM+ services role-based security.
- Manage state in JIT activated components.
- Create compensating resource managers to enable transactional support for resources lacking a resource manager.
- Use loosely coupled events to send notifications to other applications.
- Create queued components.
- Debug COM+ applications and describe common application-failure scenarios and how to solve these failures.
- Create scripts for managing deployment and administration.
- Use the enhancements provided by COM+ version 1.5 to improve the scalability, availability, and manageability of COM+ applications and use COM+ features such as services without components (SWCs), bring your own transaction (BYOT), and Phase Zero notification in applications.

AVANTUS TRAINING PTE LTD

79 Robinson Road #15-04 CPF Building Singapore 068897

Sales Hotline: (65)64163078

Email: enquiries@AvantusTraining.com

www.AvantusTraining.com

Prerequisites

Before attending this course, students must have:

- Experience building assemblies by using Microsoft Visual Basic .NET or Microsoft Visual C#.
- Experience handling database transactions in program code.

In addition, it is recommended, but not required, that students have completed:

- [Course 2349](#): Programming with the Microsoft .NET Framework (Microsoft Visual C# .NET).

Microsoft Certification exams

This course will help students prepare for these Microsoft Certification exams:

- [Exam 70-310](#): Developing XML Web Services and Server Components with Microsoft Visual Basic .NET and the Microsoft .NET Framework
- [Exam 70-320](#): Developing XML Web Services and Server Components with Microsoft Visual C# .NET and the Microsoft .NET Framework

Course Materials

The student kit includes a comprehensive workbook and other necessary materials for this class. The following software is provided in the student kit:

- Evaluation copy of Microsoft Windows XP Professional for classroom use only

Course Outline

Module 1: Introduction to COM+ Services

This module covers the evolution of applications from monolithic applications to client/server applications to component-based applications and the supporting application infrastructure that COM+ services provides. The module also covers the COM+ runtime architecture and how it uses surrogates, context, and interception to provide services to components.

Lessons

- History of Server-Based Applications
- The COM+ Runtime Architecture

After completing this module, students will be able to:

- Describe the history of server-based applications.
- Describe the COM+ runtime architecture.

Module 2: Configuring Just-in-Time Activation and Synchronization

This module describes the attributes that you can assign to components and how to write a serviced component. This module also describes how to access the object context from within code, JIT activation, synchronization, the relationship between synchronization and JIT activation, and how you can set JIT activation and synchronization for a component.

Lessons

- The .NET Enterprise Services Programming Model
- JIT Activation
- Synchronization

Lab 2: Configuring Just-in-Time Activation

- Exercise 1: Creating a Serviced Component
- Exercise 2: Using a Serviced Component

After completing this module, students will be able to:

- Use attributes to configure an assembly as a COM+ application.
- Create components that use JIT activation.
- Create components that are synchronized.

Module 3: Using ADO.NET to Work With Data

This module describes how to run a query and retrieve a result set by using ADO.NET. The module also covers how to pass parameters to a stored procedure, create typed DataSet objects, and use construction strings to specify connection information to establish a connection to a data source.

Lessons

- The ADO.NET Architecture
- Accessing a SQL Server Database

Lab 3: Using ADO.NET in a Serviced Component

- Exercise 1: Creating a New Typed Dataset
- Exercise 2: Updating the PurchasingSelect Component
- Exercise 3: Updating the OrderProcessing Component
- Exercise 4: Modifying the PlaceOrder Web Page

After completing this module, students will be able to:

- Describe the ADO.NET architecture and namespace classes.
- Use the classes provided by the SqlClient namespace to retrieve and update data from a Microsoft SQL Server 2000 database.

Module 4: Transaction Services

This module describes transaction processing, how it is implemented in .NET Enterprise Services, and how you add attributes to code to enable transaction processing.

Lessons

- Introduction to Transaction Processing
- .NET Enterprise Services Transactions

Lab 4: Using Transaction Services

- Exercise 1: Creating a Transactional Component
- Exercise 2: Updating the OrderProcessing Component
- Exercise 3: Updating the OrderApp Web Application
- Exercise 4: Testing with the OrderApproval Client

After completing this module, students will be able to:

- Describe transaction processing and how it is implemented in .NET Enterprise Services.
- Use the classes defined in the EnterpriseServices namespace to implement transaction processing.

Module 5: Securing Enterprise Applications

This module explains how to implement COM+ role-based security in serviced components by using .NET Enterprise Services.

Lessons

- Introduction to Application Security
- Implementing COM+ Role-Based Security
- Authentication and Impersonation

Lab 5: Securing Enterprise Applications

- Exercise 1: Updating the PurchasingUpdate component
- Exercise 2: Updating the OrderProcessing Component
- Exercise 3: Updating the OrderApproval Client

After completing this module, students will be able to:

- Describe the security model offered by COM+ and how it is used with other security mechanisms.
- Help protect your application by using COM+ role-based security.
- Configure authentication and impersonation levels to balance security requirements with performance and flexibility requirements.

Module 6: State Management

This module describes how to manage state in .NET Enterprise Services. It explains how to use the shared property manager (SPM) to store state, use ASP.NET applications to store application and session state, and use ASP.NET caching.

Lessons

- Introduction to State Management
- Using the Shared Property Manager
- Using ASP.NET to Store State

Lab 6: Managing Component State

- Exercise 1: Updating the OrderProcessing Component

After completing this module, students will be able to:

- Describe the need for state management and the techniques for implementing state management.
- Maintain state by using the SPM.
- Maintain state by using ASP.NET application and session state and ASP.NET caching.

Module 7: Compensating Resource Managers

This module describes the architecture of compensating resource managers (CRMs) and how to implement CRMs.

Lessons

- Introduction to Compensating Resource Managers
- Implementing Compensating Resource Managers

Lab 7: Implementing Compensating Resource Managers

- Exercise 1: Creating the OrderDocCRM Component
- Exercise 2: Updating the OrderProcessing Component

After completing this module, students will be able to:

- Describe the architecture of CRMs.
- Implement a CRM.

Module 8: Loosely Coupled Events

This module describes the architecture of Loosely Coupled Events (LCEs) and the LCE system. This module also describes how to configure and implement publishers, subscribers, and event classes.

Lessons

- Introduction to Loosely Coupled Events
- COM+ Events
- Using Loosely Coupled Events

Lab 8: Using Loosely Coupled Events

- Exercise 1: Creating an Event Class
- Exercise 2: Creating a Publisher
- Exercise 3: Creating a Subscriber

After completing this module, students will be able to:

- Describe why LCEs are needed.
- Describe the architecture of the LCE system.
- Configure the LCE system programmatically and by using the Component Services administrative tool.
- Implement publishers, subscribers, and event classes.

Module 9: Queued Components

The following topics are covered in this module:

Lessons

- Introduction to Queuing
- Developing Queued Components
- Queued Components and Transactions

Lab 9: Creating a Queued Component

- Exercise 1: Creating a Queued Component
- Exercise 2: Calling the Queued Component

After completing this module, students will be able to:

- List the advantages of using asynchronous messaging in a distributed systems environment.
- Explain the purposes of the recorder, listener, and player in the Queued Components architecture.
- List additional component design considerations introduced by asynchronous messaging.
- Install and configure a queued component in a COM+ application.
- Instantiate a queued component by using the queue and new monikers.

Module 10: Debugging COM+ Applications

In this module, you will learn how to debug applications that use .NET Enterprise Services.

Lessons

- Debugging Tools
- Common Debugging Scenarios

Lab 10: Debugging COM+ Applications

- Exercise 1: Debugging DCOM Problems
- Exercise 2: Debugging Transaction Problems
- Exercise 3: Debugging Security Problems

After completing this module, students will be able to:

- Debug COM+ applications by using tools such as the Microsoft Visual Studio debugger.
- Debug some common problems in COM+ applications.

Module 11: Deploying and Administering COM+ Applications

This module introduces the COMAdmin objects and additional techniques and tools for deploying and administering COM+ applications. It describes the advantages and drawbacks of each technique so that students can select the one most appropriate for your application.

Lessons

- Deploying a COM+ Application Built Using .NET Enterprise Services
- Using COMAdmin Objects in WSH Scripts

Lab 11: Administering COM+ Applications

- Exercise 1: Creating a COM+ Application
- Exercise 2: Adding a Role to a COM+ Application
- Exercise 3: Deleting a COM+ Application
- Exercise 4 (Optional): Exposing a COM+ Application as an XML Web Service
- Exercise 5 (Optional): Creating the OrderDispatch Service

After completing this module, students will be able to:

- Deploy a COM+ application.
- Use Microsoft Windows Script Host (WSH) scripts to do common deployment and administrative tasks, such as creating an application or adding components to an application.

Module 12: COM+ 1.5 Enhancements

This module explains how to use new COM+ version 1.5 features that help you manage, scale, and maximize the uptime of your COM+ application more efficiently. The module also explains how to use .NET Enterprise Services without having to implement serviced components.

Lessons

- Scalability and Availability Enhancements
- Manageability Enhancements
- Other COM+ Features

After completing this module, students will be able to:

- Use COM+ 1.5 features to improve the scalability and availability of COM+ applications.
- Use COM+ 1.5 features to improve the manageability of COM+ applications.
- Use COM+ features such as SWC, BYOT, and Phase Zero notification in your applications.