

Advanced Web Application Technologies with Microsoft Visual Studio 2005

Workshop GWE25440: Two days; Instructor-Led



On This Page

↓ [Introduction](#)

↓ [Audience](#)

↓ [Prerequisites](#)

↓ [Workshop Outline](#)

Introduction

This two-day instructor-led workshop provides students with the knowledge and skills to develop Microsoft ASP.NET 2.0 Web applications using Microsoft Visual Studio 2005. The workshop focuses on advanced user interfaces, Web site functionality, and implementation details using the advanced features of ASP.NET 2.0 and Visual Studio 2005.

↑ [Top of page](#)

Audience

This workshop is intended for corporate or independent software vendor (ISV) application developers who have a desire to learn more about specific technology areas in Web application development.

At Workshop Completion

After completing this workshop, students will be able to:

- Build dynamic Web applications.
- Create controls for Web applications.
- Optimize Web applications.
- Build customizable Web applications.
- Build Web Part pages and Web Parts.

↑ [Top of page](#)

Prerequisites

Before attending this workshop, students must:

- Have attended or studied Core Web Application Technologies with Visual Studio 2005, or possess equivalent knowledge and skills.
- Know how to use delegates and events.
- Know how to improve the security of .NET Framework 2.0 applications.
- Be able to use instrumentation in code.

↑ [Top of page](#)

Workshop Outline

Unit 1: Building Dynamic Web Applications

This unit introduces many different aspects of dynamic Web applications. It includes discussions on creating and configuring controls at run time. It then explains how to build dynamic globalization features into a Web application to ensure that it is localizable, including using localized resources and applying different master page layouts in response to culture and language settings. It concludes with explanations about how to enable dynamic configuration for site administrators.

Lessons

- Dynamic Control Creation
- Localization and Globalization
- Dynamic Master Pages
- Dynamic Web Configuration

Lab 1: Building Dynamic Web Applications

- Exercise 1. Dynamically Adding and Configuring Controls
- Exercise 2. Dynamically Applying Master Pages
- Exercise 3. Adding Localization Features
- Exercise 4. Dynamically Configuring Web Applications

After completing this unit, students will be able to:

- Explain dynamic control creation in ASP.NET 2.0.
- Add and configure controls dynamically.
- Explain how to incorporate globalization and localization features into Web applications.
- Add localization features to a Web application.
- Describe when and how to implement dynamic master pages.
- Apply master pages dynamically.
- Describe dynamic Web configuration scenarios.
- Dynamically configure Web applications.

Unit 2: Creating Controls for Web Applications

This unit explains how developers create different types of controls for different scenarios. The different types of controls include user controls, custom Web server controls, composite Web server controls, and templated controls.

Lessons

- User Controls
- Custom Web Server Controls
- Composite Web Server Controls
- Templated Controls

Lab 2: Creating Controls for Web Applications

- Exercise 1. Creating User Controls
- Exercise 2. Creating Custom Web Server Controls
- Exercise 3. Creating Composite Web Server Controls
- Exercise 4. Creating Templated Controls

After completing this unit, students will be able to:

- Describe user controls and the underlying enabling technologies.
- Create user controls.
- Describe custom Web server controls and the underlying enabling technologies.
- Create Web server controls.
- Describe composite controls and how composite controls are created.
- Create composite Web server controls.
- Describe templated controls and the interfaces that enable their implementation.
- Create templated controls.

Unit 3: Optimizing Web Application Performance

This unit introduces topics that will help you improve the performance of Web applications. It describes how the Page Scripting Object Model can help reduce the number of round trips for communication between the server and the browser, and then explains how tracing and instrumentation can be used to monitor and, therefore, improve the performance of a Web application. The unit discusses how caching and asynchronous processing can help increase Web application performance; it then highlights some considerations that developers must address if the Web application is to be deployed in a Web farm environment.

Lessons

- The Page Scripting Object Model
- Tracing and Instrumentation in Web Applications
- ASP.NET 2.0 Caching Techniques
- Asynchronous Processing in Web Applications
- Web Farm Development Considerations

Lab 3: Optimizing Web Application Performance

- Exercise 1. Accessing the Page Scripting Object Model
- Exercise 2. Implementing ASP.NET Caching Techniques
- Exercise 3. Implementing Tracing and Instrumentation Techniques in Web Applications
- Exercise 4. Implementing Asynchronous Processing in Web Applications

After completing this unit, students will be able to:

- Describe the Page Scripting Object Model.
- Access Page Scripting Object Model functionality.
- Explain how to use tracing and instrumentation to monitor and improve the performance of a Web application.
- Implement tracing and instrumentation in Web applications.
- Describe ASP.NET 2.0 caching techniques.
- Implement ASP.NET 2.0 caching techniques.
- Explain how asynchronous processing can lead to improved performance for Web applications.
- Implement asynchronous processing in Web applications.
- Describe strategies for dealing with session state management issues when deploying Web applications in a Web farm environment.
- Develop Web applications for Web farm environments.

Unit 4: Implementing Personalization and Themes in Web Applications

This unit introduces building customizable functionality into a Web application by adding personalization support. It discusses using the personalization features of ASP.NET 2.0 to provide this functionality. In addition, it discusses applying themes to Web applications and allowing users to choose color schemes to personalize their experience in using the Web application. It concludes by explaining how to include features that enable users to personalize themes.

Lessons

- ASP.NET 2.0 Personalization Features
- Theme Support in ASP.NET 2.0

Lab 4: Implementing Personalization and Themes in Web Applications

- Exercise 1. Configuring Personalization
- Exercise 2. Implementing Personalization Functionality
- Exercise 3. Adding Themes to the Web Application
- Exercise 4. Implementing Personalized Themes

After completing this unit, students will be able to:

- Describe the personalization features provided by ASP.NET 2.0.
- Describe ASP.NET 2.0 theme support.
- Configure personalization for a Web application.
- Implement personalization features.
- Add themes to a Web application.
- Implement customizable themes.

Unit 5: Building Web Part Pages and Web Parts

This unit introduces the concept of a Web part, and describes how it is used in portal pages and other scenarios. It introduces the concept of a Web part page, and discusses how a Web part page contains some Web parts that provide the user interface, along with other controls that manage the Web part infrastructure. Additionally, it introduces the advanced features of connected Web parts and discusses scenarios where they are typically used.

Lessons

- What Is a Web Part?
- What Is a Web Part Page?
- Connected Web Parts

Lab 5: Building Web Part Pages and Web Parts

- Exercise 1. Creating a Web Part Page
- Exercise 2. Creating a Web Part
- Exercise 3. Creating Connected Web Parts

After completing this unit, students will be able to:

- Describe what a Web Part is and the purpose of Web Parts.
- Describe the components of a Web Part page and identify scenarios when Web Part pages are useful features of Web applications.
- Describe the more advanced features of Web Parts, including connections between Web Parts.
- Create Web Part pages.
- Create Web Parts.

- Create connected Web Parts.