

EFFECTIVE SOFTWARE CONFIGURATION MANAGEMENT USING TEAM SYSTEM 2008



Course SCM08: Three days; Instructor-Led Course Syllabus

INTRODUCTION

This three-day, instructor-led course provides students with the knowledge and skills to effectively identify, manage, and track changes to software. Intended for teams working in parallel, this course focuses on using Visual Studio Team System 2008 to maximize productivity while minimizing mistakes to reduce the amount of effort expended when changes must be made.

AUDIENCE

This course is intended for current software development professionals who want to work more effectively in parallel. Some students may be part of distributed teams who work on the same project as others but without ideal communication. Other students may be working on multiple versions of the same application at the same time. Regardless of the situation, students will be able to learn and get hands-on experience with all of the Software Configuration Management related features of Visual Studio Team System 2008.

AT COURSE COMPLETION

After attending this course, students will be able to:

- Understand Software Configuration Management (SCM)
- Understand Team System's support of SCM
- Identify and control changes using Team System
- Ensure changes are properly implemented using Team System
- Use Team System to report changes to others who may have an interest
- Protect Visual Studio projects by placing them under Team Foundation Version Control
- Work with Version Control in a multi-user environment
- Enforce quality using check-in constraints
- Automate the software build process using Team Build
- Ensure quality using build verification tests

PREREQUISITES

Before attending this course, students should have experience working on a team-based software development project and be familiar with their organization's Software Development Life Cycle.

- Have familiarity with one or more software development methodologies
- Have used Visual Studio (any version)
- Understand the basic foundations of .NET
- Understand basic C# .NET code (all source code will be provided)

COURSE OUTLINE

Module 1: Introduction to Software Configuration Management

This module introduces Software Configuration Management and the related support in Team System, from both a project management and software engineering point of view.

Lessons

- Introduction to Software Configuration Management
- Team System support of SCM
- Team System support for identifying change
- Team System support for controlling change
- Team System support for ensuring changes are being properly implemented
- Team System support for reporting changes

Lab Exercises

- Review the SCM environment and project scenario
- Meet the team and review their security permissions
- Review the architecture and related projects

Module 2: Identifying Change – Team Projects

This module introduces team projects, process templates, work items, and version control options in Team Foundation Server 2008. Each of these can be used to identify and track the changes pertaining to a software development project.

Lessons

- Introduction to team projects
- Planning and creating team projects
- Configuring team projects
- Configuring security, areas, iterations, version control settings
- Methodologies and process templates
- Managing work items
- Team Explorer and other client applications
- Managing the project portal

Lab Exercises

- Create and configure a team project
- Secure a team project and its SharePoint project portal
- Create and review work items using various client applications:
- Use Team Explorer, Microsoft Excel, Microsoft Project, and the Web Access Power Tool
- Upload documents to the project portal
- Explore the process template
- Explore the widget: Process Editor (optional)

Module 3: Controlling Change – Team Foundation Version Control

This module is the first of three to focus on Team Foundation Version Control. It introduces the primary version control capabilities and explains how to plan and setup version control for team projects in order to control change in a parallel development environment.

Lessons

- Introduction to controlling change
- Using version control as communication
- Planning version control, creating and managing workspaces
- Integration with Visual Studio 2008, alternate clients using MSSCCI provider
- Get, check-out, pending changes, check-in, changesets, undo
- Viewing history, annotating files, comparing revisions
- Migrating from VSS and other SCC systems (optional)

Lab Exercises

- Configure Team Foundation Version Control
- Create and manage workspaces
- Setup SCM folder structures
- Work with folders and files – get, check-out, check-in, view pending, view history, etc.
- Explore version controlled artifacts using the Web Access Power Tool
- Explore the widget: MSSCCI provider (optional)
- Explore the widget: Workspace Sidekick (optional)

Module 4: Controlling Change – Parallel Development

This module continues the discussion of Team Foundation Version Control, focusing on the topics related to multiple users working on multiple, sometimes simultaneous, projects and project versions in an agile environment. The topics in this module cover branching, merging, shelving, and detecting/mitigating conflicts when they occur.

Lessons

- Introduction to parallel development, terminology
- Parallel development scenarios, locking models
- Branching, merging, and promotion modeling
- Branching patterns, conflict detection and resolution
- Shelving, unshelving, achieving peer review using shelving
- Securing version control, specifications
- Using Team Foundation Server 2008 Power Tools

Lab Exercises

- Understand and resolve conflicts in a multi-user environment
- Branch code, manage branches, merge changes between branches
- Use the command-line tool to perform a baseless merge
- Secure artifacts in version control
- Shelf and unshelf changes
- Explore the widget: alternate merge tool (optional)

Module 5: Ensuring Changes are Properly Implemented – Check-In Policies

This module completes the discussion of Team Foundation Version Control, focusing on ensuring the quality of the checked-in artifacts. The module will discuss the topics related to using the standard policies to run tests, code analysis, and verify work item association, as well as creating and using custom check-in policies.

Lessons

- Ensuring changes are properly implemented
- Configuring check-in policies and notes
- Using Team Foundation Server 2008 Power Tools policy-pack
- Understanding policy failure, overriding, and managing overrides
- Creating, deploying, and testing a custom check-in policy

Lab Exercises

- Configure the work item association check-in policy
- Configure the code analysis check-in policy
- Migrate code analysis policy settings to Visual Studio projects
- Create a unit test and manage test lists
- Configure a unit testing check-in policy
- Use the Custom Path policy to scope other check-in policies
- Create, deploy, and test a custom check-in policy (optional)
- Explore the widget: Policy Override Notification Tool (optional)

Module 6: Ensuring Changes are Properly Implemented – Team Build

This module covers the Team Build features of Team Foundation Server, including the architecture and execution process. Topics include the creating, configuring, and execution of build scripts, as well as how to customize the scripts and extend the notification services to execute builds and associated tests automatically.

Lessons

- Overview of Team Foundation Build
- The Team Build process
- Creating and queuing build definitions
- Running tests and computing code coverage
- Automating Team Build
- Continuous integration
- Customizing MSBuild, targets, tasks, and creating custom tasks

Lab Exercises

- Create a Team Build definition
- Queue (execute) the automated build in various ways
- Schedule an automated build
- Ensure a quality build by enabling code analysis
- Enable build notifications
- Configuring continuous integration
- Customize Team Build definition and create custom MSBuild tasks

Module 7: Reporting Change – Queries, Reports, and Traceability

This module covers the querying, reporting, and notification capabilities of Team System. Topics include querying work items and their audit history, executing various reports, creating custom reports, setting up email notifications, and the overall traceability of work items through check-ins and finally to a build.

Lessons

- Methods to track and report changes
- Understanding quality metrics
- Using queries to track work items
- Auditing work items and achieving traceability
- Running reports and creating custom reports
- Working with the warehouse controller service
- Understanding and configuring notifications

Lab Exercises

- Query work items from team explorer
- View work item audit history and trace work items to changesets
- Use WIQL to query work items
- Use Fiddler and SQL Server Profiler to understand WIQL execution
- Use the Visual Studio SDK to learn about WIQL execution
- Query work items from Microsoft Office and Internet Explorer
- Create an Excel Pivot table and chart work item data
- Run reports from the project portal, Report Manager, and Team Explorer
- Explore the Team Foundation Server data warehouse
- Create and execute a custom report
- Explore the widget: Warehouse Status Update Tool (optional)

Module 8: SCM Patterns and Best Practices

This module describes some common patterns you can use when implementing software configuration management, especially with regards to branching and merging models. Several Team System and SCM best practices will also be highlighted in this module.

Lessons

- Terminology and nomenclature
- Common scenarios and SCM patterns for those scenarios
- Best Practice: when to create a new team project
- Best Practice: when to branch and merge
- Best Practice: managing workspaces
- Best Practice: balance speed and quality
- Best Practice: balance autonomous work and integration
- SCM Implementation in an agile software development
- Various How-To topics
- References

Lab Exercises

- None

Course Designer

This course was designed by Richard Hundhausen of Accentient, Inc. Richard is a Visual Studio Team System MVP and Microsoft Regional Director, as well as an experienced developer and trainer.

For more information, visit www.accentient.com