

# TEAM FOUNDATION SERVER 2008 IN ACTION

Jamil Azher



# contents

---

*preface* xiii  
*acknowledgments* xvi  
*about this book* xvii

---

## **PART 1 EXPLORING TEAM FOUNDATION SERVER.....1**

---

- 1** *TFS and the practice of software development* 3
  - 1.1 Background on software development processes 5
  - 1.2 TFS architecture 6
    - Client tier* 7 ▪ *Application tier* 7 ▪ *Data tier* 7
  - 1.3 Major features 8
    - Work item tracking* 8 ▪ *Version control* 9 ▪ *Build automation* 10 ▪ *Project portal* 12 ▪ *Reporting* 13
  - 1.4 Summary 14
  - 1.5 References 15

- 2 Exploring the changes in TFS 2008 16**
- 2.1 Team Build 18
    - Topology and security 18* ▪ *Build agents and team projects 20*
    - Understanding the structure of the build directory 21* ▪ *Creating a new build definition 22* ▪ *Launching and queuing builds 28*
    - Build definitions and security 29* ▪ *Running parallel builds 29*
    - Using the Build Explorer 30* ▪ *Managing build qualities 32*
    - Generating incremental builds 33* ▪ *Exploring the Team Build object model 33* ▪ *A sample program for queuing builds 35*
    - Determining which build contains a particular fix 37*
  - 2.2 Team Foundation version control 40
    - Get the latest on check-out 40* ▪ *Working offline 43*
    - Folder comparison 44* ▪ *Annotate 45*
  - 2.3 Summary 47
- 3 Introducing VSTS 2008 Database Edition 48**
- 3.1 A quick tour of DB Pro 50
    - Creating a database project 51* ▪ *Making changes offline 54*
    - Deploying modified schema to a test database 56* ▪ *Generating test data 56* ▪ *Creating unit tests 59* ▪ *Deploying the modified schema to production database 61*
  - 3.2 Building database projects 64
    - Using MSBuild to build database projects 75* ▪ *Using Team Build to build database projects 76*
  - 3.3 Technical review: creating a custom data generator 79
  - 3.4 Summary 83

## **PART 2 DIVING DEEP INTO VERSION CONTROL AND TEAM BUILD.....85**

---

- 4 Understanding branching in version control 87**
- 4.1 Benefits of using a version control system 89
  - 4.2 Company types and branching issues 90
    - Small companies 90* ▪ *Large companies 90* ▪ *Large companies with distributed TFS 90*
  - 4.3 Branching models 92
    - The code promotion model 92* ▪ *The codeline-per-release model 94*
    - The branch-by-release model 96* ▪ *A branching model for large projects 97* ▪ *A remote branch model for distributed teams 99*

- 4.4 Branches versus labels 100
  - Identifying changes after creating a label* 103
- 4.5 Branches versus folders 106
- 4.6 Sharing code between team projects 108
- 4.7 Branches and builds 111
- 4.8 Using Team Foundation Sidekicks 115
- 4.9 Summary 117
- 4.10 References 118

## **5 Understanding branch policies 119**

- 5.1 Policies and branches 120
  - Access control policies* 120 ▪ *Check-in policies* 120
- 5.2 Determining access control policies 121
  - Understanding locks* 123 ▪ *Using the TFSSecurity tool* 123
- 5.3 Using the Custom Path policy 124
- 5.4 Creating a custom policy 125
  - Understanding the core interfaces* 127 ▪ *Technical implementation* 128 ▪ *Installing custom policies* 139
- 5.5 Summary 140

## **6 Understanding merging in version control 141**

- 6.1 Viewing merge history 142
- 6.2 Viewing changes 144
- 6.3 Comparing contents of two branches 145
- 6.4 Determining merge candidates and conflicts 147
- 6.5 Resolving merge conflicts 148
- 6.6 Rolling back a merge 150
- 6.7 Understanding baseless merges 152
- 6.8 Notifying developers about concurrent changes 154
  - The TFS event engine* 154 ▪ *Technical review: implementing an event receiver web service* 156
- 6.9 Summary 159

## **7 Understanding Team Build 160**

- 7.1 Company types and build issues 162
  - Small companies* 162 ▪ *Large companies* 163 ▪ *Large companies with distributed TFS* 163

- 7.2 An overview of Team Build 164
    - Architecture* 164
    - *The build project file* 165
    - *Target execution order* 165
    - *Customizing targets* 170
    - *Detecting the reason for build failure* 171
    - *Build directories* 171
    - *Coupling between TFS and build machines* 173
    - *Associating changesets and related work items* 174
  - 7.3 Creating builds from labeled source files 176
    - Creating a new build from a label* 176
    - *Reproducing a previous build from a label* 177
  - 7.4 Deploying ClickOnce 182
    - Configuring ClickOnce* 183
    - *Extending Team Build to incorporate ClickOnce* 185
    - *Creating a custom task to configure deployment files* 186
    - *Configuring ClickOnce using Team Build tasks* 188
  - 7.5 A source code submission service 189
    - Step 1—Create a shelve set for code submission* 189
    - Step 2—Submit change information via submission web site* 190
    - Step 3—Process requests using the submission service* 191
  - 7.6 Technical review: creating the source code submission service 192
    - Creating a Team Build project and downloading baseline source files* 192
    - *Downloading the shelve set* 193
    - *Running prebuild checks* 195
    - *Launching the build process* 196
    - *Evaluating build results* 198
  - 7.7 Desktop builds 199
  - 7.8 TFS and distributed builds 200
    - Scenario: central build machine* 201
    - *Scenario: distributed build machines* 202
  - 7.9 Summary 203
  - 7.10 References 204
- 
- 8** ***Versioning assemblies using Team Build*** 205
    - 8.1 Assembly versioning background 206
    - 8.2 Versioning Windows applications 208
    - 8.3 Customizing Team Build to support assembly numbering 209
      - Technical approach* 210
      - *Technical Implementation* 216

- 8.4 Versioning web applications 222
  - Building web sites* 222 ▪ *Building web sites using VS 2005 Web Deployment Projects* 224 ▪ *Strategy for autoincrementing build numbers* 226 ▪ *Modifying the team project file* 227
- 8.5 Summary 228

## PART 3 ADMINISTERING AND CUSTOMIZING TFS .....229

### 9 *Configuring and extending TFS* 231

- 9.1 Company types and TFS configuration issues 232
  - Small companies* 232 ▪ *Large companies* 232 ▪ *Large companies with distributed TFS* 233
- 9.2 Organizing team projects 233
  - Work items* 233 ▪ *Version control* 234 ▪ *Reports* 235
  - Project portal* 235 ▪ *Process model* 236 ▪ *Security* 236
  - General recommendations for organizing team projects* 238
- 9.3 Limitations and considerations when creating team projects 238
  - System limitations* 238 ▪ *Moving team projects* 239
- 9.4 Performance and high availability 241
  - Load balancing* 241 ▪ *Reducing load on the application-tier machine* 242 ▪ *Reducing load on the data-tier machine* 243 ▪ *64-bit processor support* 243 ▪ *High availability and disaster recovery* 243
- 9.5 Using custom controls in work item forms 244
  - Creating custom controls* 245 ▪ *Technical review: creating custom controls* 247
- 9.6 Creating and using KPIs for TFS 253
  - Step 1—Creating a business intelligence project* 253
  - Step 2—Creating calculated members* 255
  - Step 3—Constructing KPIs* 257
  - Step 4—Viewing KPIs* 259
- 9.7 Summary 267

<b>10</b>	<b><i>Using workflow with TFS</i></b>	<b>268</b>
10.1	Company types and workflow issues	269
	<i>Small companies</i>	270
	<i>Large companies</i>	270
	<i>Large companies with distributed TFS</i>	271
10.2	Built-in workflow capabilities in TFS	271
10.3	An overview of Windows Workflow Foundation	272
	<i>Layers in WF</i>	273
	<i>Creating WF applications</i>	275
10.4	Creating a central build from distributed TFS machines	276
10.5	Technical review: using WF to manage a distributed build process	278
	<i>Step 1—Downloading source files from the reference project</i>	279
	<i>Step 2—Handling failed download of the reference project</i>	281
	<i>Step 3—Handling successful download of the reference project</i>	282
	<i>Step 4—Downloading source files from the primary project</i>	284
	<i>Step 5—Handling failed download of the primary project</i>	284
	<i>Step 6—Handling successful download of the primary project</i>	286
	<i>Step 7—Handling outgoing and incoming emails</i>	292
10.6	Enabling workflow persistence	303
10.7	Enabling workflow tracking	308
10.8	Summary	311
	<i>index</i>	313