

Dependency Injection in .NET

MARK SEEMANN



MANNING
SHELTER ISLAND

contents

<i>foreword</i>	<i>xvii</i>
<i>preface</i>	<i>xix</i>
<i>acknowledgments</i>	<i>xxi</i>
<i>about this book</i>	<i>xxiv</i>
<i>about the cover illustration</i>	<i>xxix</i>

PART 1. PUTTING DEPENDENCY INJECTION ON THE MAP ...1

"f A Dependency Injection tasting menu 3

1.1	Writing maintainable code	5
	<i>Unlearning DI</i>	<i>5</i>
	<i>Understanding the purpose of DI</i>	<i>8</i>
1.2	Hello DI	13
	<i>Hello DI code</i>	<i>13</i>
	<i>Benefits of DI</i>	<i>15</i>
1.3	What to inject and what not to inject	22
	<i>Seams</i>	<i>22</i>
	<i>Stable Dependencies</i>	<i>23</i>
	<i>Volatile Dependencies</i>	<i>23</i>
1.4	DI scope	24
	<i>Object Composition</i>	<i>25</i>
	<i>Object Lifetime</i>	<i>26</i>
	<i>Interception</i>	<i>26</i>
	<i>DI in three dimensions</i>	<i>27</i>
1.5	Summary	28

A comprehensive example 29

2.1 Doing it wrong 30

Building a tightly coupled application 31*Smoke test* 36 * *Evaluation* 37 • *Analysis* 39

2.2 Doing it right 41

Rebuilding the commerce application 43 • *Analyzing the loosely coupled implementation* 51

2.3 Expanding the sample application 53

Architecture 53 • *Basket feature* 54

2.4 Summary 57

DI Containers 58

3.1 Introducing DI Containers 61

Hello container 62 • *Auto-wiring* 64

3.2 Configuring DI Containers 67

Configuring containers with XML 68 • *Configuring containers with code* 70 * *Configuring containers by convention* 72

3.3 DI Container patterns 75

Composition Root 75 • *Register Resolve Release* 81

3.4 DI Container landscape 87

Selecting a DI Container 87 • *Microsoft and DI* 89

3.5 Summary 91

PART 2 DI CATALOG..... 93

A DI patterns 95

* 4.1 Constructor Injection 98

How it works 98 * *When to use it* 99 • *Known use* 100*Example: Adding a currency provider to the shopping basket* 101*Related patterns* 103

4.2 Property Injection 104

How it works 104 • *When to use it* 105 • *Known use* 107*Example: Defining a currency profile service for the**BasketController* 108 * *Related patterns* 110

4.3 Method Injection 111

How it works 111 • *When to use it* 112 • *Known use* 113*Example: Converting baskets* 114 • *Related patterns* 117

- 4.4 Ambient Context 118
 - How it works 118* When to use it 120* Known use 123*
 - Example: Caching Currency 123* Related patterns 130*
- 4.5 Summary 131

DI anti-patterns 133

- 5.1 Control Freak 136
 - Example: newing up Dependencies 136 * Example: Factory 137*
 - Analysis 143*
- 5.2 Bastard Injection 144
 - Example: ProductService with Foreign Default 144*
 - Analysis 146*
- 5.3 Constrained Construction 149
 - Example: late-binding ProductRepository 149*
 - Analysis 151*
- 5.4 Service Locator 154
 - Example: ProductService using a Service Locator 156*
 - Analysis 157*
- 5.5 Summary 160

DI refactorings 162

- 6.1 Mapping runtime values to Abstractions 163
 - Abstractions with runtime Dependencies 164*
 - Example: selecting a routing algorithm 166*
 - Example: using a CurrencyProvider 168*
- 6.2 Working with short-lived Dependencies 170
 - Closing connections through Abstractions 170*
 - Example: invoking a product-management service 173*
- 6.3 Resolving cyclic Dependencies 175
 - Addressing Dependency cycles 176*
 - Example: composing a window 178*
- 6.4 Dealing with Constructor Over-injection 182
 - Recognizing and addressing Constructor Over-injection 182*
 - Example: refactoring order reception 185*
- 6.5 Monitoring coupling 188
 - Unit-testing coupling 189* Integration-testing coupling 191*
 - Using NDepend to monitor coupling 193*
- 6.6 Summary 195

Object Composition 199

- 7.1 Composing console applications 202
Example: updating currencies 202
- 7.2 Composing ASP.NET MVC applications 206
ASP.NET MVC extensibility 206 Example: implementing
CommerceControllerFactory 208*
- 7.3 Composing WCF applications 210
*WCF extensibility 211 * Example: wiring up
a product-management service 212*
- 7.4 Composing WPF applications 219
*WPF Composition 219 * Example: wiring up
a product-management rich client 220*
- 7.5 Composing ASP.NET applications 224
*ASP.NET composition 224 * Example: wiring up
a CampaignPresenter 225*
- 7.6 Composing PowerShell cmdlets 230
Example: composing basket-management cmdlets 231
- 7.7 Summary 235

Object Lifetime 236

- 8.1 Managing Dependency Lifetime 239
*Introducing Lifetime Management 239
Managing lifetime with a container 242*
- 8.2 Working with disposable Dependencies 247
*Consuming disposable Dependencies 248
Managing disposable Dependencies 251*
- 8.3 Lifestyle catalog 255
*Singleton 255 * Transient 258 * Per Graph 259
WebRequest Context 261 * Pooled 266 * Other lifestyles 271*
- 8.4 Summary 273

Interception 275

- 9.1 Introducing Interception 277
Example: implementing auditing 277 Patterns and principles
for Interception 281*

- 9.2 Implementing Cross-Cutting Concerns 285
 - Intercepting with a Circuit Breaker* 286
 - Handling exceptions* 292 • *Adding security* 293
- 9.3 Declaring aspects 295
 - Using attributes to declare aspects* 296
 - Applying dynamic Interception* 300
 - Example: intercepting with Windsor* 303
- 9.4 Summary 308

PART 4 DI CONTAINERS.....311

10 Castle Windsor 313

- 10.1 Introducing Castle Windsor 314
 - Resolving objects* 315 * *Configuring the container* 317
 - Packaging configuration* 322
- 10.2 Managing lifetime 323
 - Configuring lifestyle* 324 * *Using advanced lifestyles* 325
 - Developing a custom lifestyle* 327
- 10.3 Working with multiple components 333
 - Selecting among multiple candidates* 333
 - Wiring sequences* 336 * *Wiring Decorators* 339
- 10.4 Configuring difficult APIs 341
 - Configuring primitive Dependencies* 341
 - Registering components with code blocks* 343
 - Wiring with Property Injection* 344
- 10.5 Summary 345

11 StructureMap 347

- 11.1 Introducing StructureMap 348
 - Resolving objects* 350 * *Configuring the container* 352
 - Packaging configuration* 358
- 11.2 Managing lifetime 361
 - Configuring lifestyles* 362 * *Developing a custom lifestyle* 364
- 11.3 Working with multiple components 370
 - Selecting among multiple candidates* 371
 - Wiring sequences* 374 * *Wiring Decorators* 377

- 11.4 Configuring difficult APIs 380
*Configuring primitive Dependencies 380 * Creating objects with code blocks 381 • Wiring with Property Injection 382*
- 11.5 Summary 383

12 *Spring.NET* 385

- 12.1 Introducing Spring.NET 386
*Resolving objects 387 * Configuring the container 389 Loading XML 393*
- 12.2 Managing lifetime 397
Configuring object scopes 398
- 12.3 Working with multiple components 399
*Selecting among multiple candidates 400 Wiring sequences 402 * Wiring Decorators 405 Creating Interceptors 407*
- 12 A Configuring difficult APIs 412
*Configuring primitive Dependencies 412 * Configuring static factories 413* Wiring with Property Injection 414*
- 12.5 Summary 416

13 *Autofac* 417

- 13.1 Introducing Autofac 418
*Resolving objects 420 * Configuring the ContainerBuilder 422 Packaging configuration 427*
- 13.2 Managing lifetime 429
Configuring instance scope 430
- 13.3 Working with multiple components 433
*Selecting among multiple candidates 434 Wiring sequences 438 * Wiring Decorators 440*
- 13.4 Registering difficult APIs 442
*Configuring primitive Dependencies 443 * Registering objects with code blocks 444 * Wiring with Property Injection 445*
- 13.5 Summary 447

14 *Unity* 448

- 14.1 Introducing Unity 450
*Resolving objects 451 * Configuring the container 453 Packaging configuration 458*

- 14.2 Managing lifetime 459**
 - Configuring lifetime 460 * Developing a custom lifetime 464*
- 14.3 Working with multiple components 473**
 - Selecting among multiple candidates 473*
 - Wiring sequences 476 * Wiring Decorators 479*
 - Creating Interceptors 481*
- 14.4 Configuring difficult APIs 486**
 - Configuring primitive Dependencies 486*
 - Registering components with code blocks 487*
 - Wiring with Property Injection 489*
- 14.5 Summary 490**
- MEF 492**
 - 15.1 Introducing MEF 495**
 - Resolving objects 496 * Defining imports and exports 499*
 - Working with catalogs 504*
 - 15.2 Managing lifetime 509**
 - Declaring creation policy 509 * Releasing objects 511*
 - 15.3 Working with multiple components 513**
 - Selecting among multiple candidates 513*
 - Wiring sequences 516 " Wiring Decorators 519*
 - 15.4 Composing difficult APIs 521**
 - Compositing primitive parts 521 * Composing parts with non-public constructors 522 * Wiring with Property Injection 523*
 - 15.5 Summary 524**
 - resources 526*
 - glossary 531*
 - index 535*