2.4 Modularity
  2.4.1 Exporting a Package 19
  2.4.2 Importing a Package 20
  2.4.3 Requiring a Bundle 20
  2.4.4 Enforcing Modularity 21
2.5 Modular Design Concepts 21
2.6 Lifecycle 22
2.7 Collaboration
  2.7.1 Services 24
  2.7.2 Extensions and Extension Points 25
2.8 The OSGi Framework 26
2.9 Security 27
2.10 OSGi Framework Implementations 27
2.11 Summary 28

Part II  O  OSGi by Example 29

Chapter 3 Tutorial Introduction
  3.1 What Is Toast? 31
  3.2 The Evolution of Toast 34
  3.3 Development Environment Installation 36
  3.4 Sample Code
    3.4.1 Moving from Chapter to Chapter 37
    3.4.2 Comparing 38
  3.5 Target Platform Setup
    3.5.1 The Predefined Target 40
    3.5.2 Defining Target Platforms 42
  3.6 Learning by Example 46
  3.7 Summary 48

Chapter 4 Hello, Toast
  4.1 A Simple Scenario
    4.1.1 The Project 50
    4.1.2 Gps 51
Chapter 9 Packaging

9.1 Defining a Toast Product
   9.1.1 Creating a Product Configuration
   9.1.2 The Overview Page
   9.1.3 The Dependencies Page
   9.1.4 The Configuration Page
   9.1.5 The Launching Page
   9.1.6 Running the Product
   9.1.7 Productizing the Client

9.2 Exporting Toast

9.3 Packaging for Other Platforms

9.4 Getting Serious about Component Definition
   9.4.1 Versions and Version Ranges
   9.4.2 Exporting Packages and Friendship

9.5 Summary

Chapter 10 Pluggable Services

10.1 Separating Interface from Implementation
   10.1.1 Separating the Fake Airbag from Its Interface
   10.1.2 Separating the Fake GPS from Its Interface
   10.1.3 Regression Testing

10.2 Device Simulation
   10.2.1 Concepts
   10.2.2 The Device Simulator Framework

10.3 Simulated Devices as Pluggable Services
   10.3.1 The Simulated Airbag
   10.3.2 The Simulated GPS

10.4 Running with Simulated Devices

10.5 Summary
Chapter 11 Extensible User Interface

11.1 Crust
   11.1.1 Crust Shell 174
   11.1.2 Crust Widgets 175

11.2 Emergency
   11.2.1 Making the User Interface Pluggable 176
   11.2.2 Refactoring the Emergency Domain Logic 176
   11.2.3 The Emergency User Interface 178
   11.2.4 Running the User Interface 179

11.3 Climate and Audio
   11.3.1 Climate and Audio Devices 181
   11.3.2 Climate and Audio Screens 183
   11.3.3 Running the User Interface 183

11.4 The OSGi Application Model 184

11.5 Navigation and Mapping
   11.5.1 Google Earth Integration 187
   11.5.2 Mapping Support 191
   11.5.3 Application Extensibility and Navigation Support 191
   11.5.4 Running the User Interface 192

11.6 Summary 195

Chapter 12 Dynamic Configuration

12.1 The Tracking Scenario 197
12.2 Installing the Tracking Code 198
   12.2.1 The Core Tracking Bundle 198
   12.2.2 The Back End Tracking Bundle 199
   12.2.3 The Client Tracking Bundle 199

12.3 Running the Basic Tracking Scenario 201

12.4 Configuration 201
   12.4.1 OSGi’s Configuration Admin 201
   12.4.2 The Client Tracking Bundle 201
<table>
<thead>
<tr>
<th>Content</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.4.3 Running Configurable Toast</td>
<td>204</td>
</tr>
<tr>
<td>12.4.4 Running with Persistent Configuration</td>
<td>205</td>
</tr>
<tr>
<td>12.5 Summary</td>
<td>205</td>
</tr>
<tr>
<td>Chapter 13 Web Portal</td>
<td>207</td>
</tr>
<tr>
<td>13.1 Portal</td>
<td>207</td>
</tr>
<tr>
<td>13.2 The PortalServlet</td>
<td>208</td>
</tr>
<tr>
<td>13.3 Action Lookup Using Services</td>
<td>210</td>
</tr>
<tr>
<td>13.4 Declaring a Portal Action</td>
<td>213</td>
</tr>
<tr>
<td>13.5 Whiteboard Pros and Cons</td>
<td>215</td>
</tr>
<tr>
<td>13.6 Summary</td>
<td>216</td>
</tr>
<tr>
<td>Chapter 14 System Deployment with p2</td>
<td>217</td>
</tr>
<tr>
<td>14.1 Introduction to Equinox p2</td>
<td>217</td>
</tr>
<tr>
<td>14.1.1 Architecture</td>
<td>218</td>
</tr>
<tr>
<td>14.1.2 p2 Metadata—Installable Units</td>
<td>219</td>
</tr>
<tr>
<td>14.1.3 Artifacts</td>
<td>220</td>
</tr>
<tr>
<td>14.1.4 Repositories</td>
<td>220</td>
</tr>
<tr>
<td>14.1.5 Profiles</td>
<td>220</td>
</tr>
<tr>
<td>14.1.6 Director</td>
<td>221</td>
</tr>
<tr>
<td>14.1.7 Engine</td>
<td>221</td>
</tr>
<tr>
<td>14.2 Refining the Toast Structure</td>
<td>221</td>
</tr>
<tr>
<td>14.2.1 Defining Products Using Features</td>
<td>222</td>
</tr>
<tr>
<td>14.2.2 Back End Features</td>
<td>222</td>
</tr>
<tr>
<td>14.2.3 Client Features</td>
<td>225</td>
</tr>
<tr>
<td>14.2.4 Restructuring Summary</td>
<td>228</td>
</tr>
<tr>
<td>14.3 Writing a Provisioner</td>
<td>229</td>
</tr>
<tr>
<td>14.3.1 The Provisioner</td>
<td>229</td>
</tr>
<tr>
<td>14.3.2 Configuring the Back End</td>
<td>233</td>
</tr>
<tr>
<td>14.3.3 Back End Summary</td>
<td>233</td>
</tr>
<tr>
<td>14.4 Adding a Deployment Web UI</td>
<td>233</td>
</tr>
<tr>
<td>14.4.1 The Create Action</td>
<td>234</td>
</tr>
<tr>
<td>14.4.2 The Manage Action</td>
<td>235</td>
</tr>
</tbody>
</table>
14.4.3 The Install and Uninstall Actions 235
14.4.4 Installing the Provisioning UI 235
14.5 Exporting, Running, and Provisioning 235
  14.5.1Populating a p2 Repository 236
  14.5.2 Running the Toast Back End 238
  14.5.3 Creating and Provisioning Vehicles 240
14.6 Client-Side Dynamic Deployment 241
14.7 Summary 242

Part III 〇 Deep Dives 245

Chapter 15  Declarative Services 247
  15.1 The Declarative Services Model 247
  15.2 Common Scenarios 248
    15.2.1 The Simplest Component 249
    15.2.2 Referencing Services 250
    15.2.3 Providing Services 253
    15.2.4 Referencing and Providing Services 254
    15.2.5 Immediate Components 255
    15.2.6 The Whiteboard Pattern 256
    15.2.7 Factory Components 262
  15.3 Launching and Debugging DS Applications 269
  15.4 PDE Tooling 270
  15.5 Summary 273

Chapter 16  Extensions 275
  16.1 The Extension Registry 275
  16.2 Extension Points 278
  16.3 Extensions 280
  16.4 Advanced Extension Topics 281
    16.4.1 Contribution IDs 281
    16.4.2 Named and Anonymous Extensions 282
    16.4.3 Extension Factories 282
16.5 Extension Registry Lifecycle 283
16.6 Dynamic Extension Scenarios 284
   16.6.1 Scenario 1: No Caching 285
   16.6.2 Scenario 2: Extension Caching 285
   16.6.3 Scenario 3: Object Caching 287
16.7 Services and Extensions 290
   16.7.1 Integrating Services and Extensions 291
16.8 Extension Registry Myths 293
16.9 Summary 293

Chapter 17 Logging 295
17.1 The Log Service Specification 295
   17.1.1 Logging Levels 296
   17.1.2 Writing to the Log 296
   17.1.3 Reading the Log 297
   17.1.4 Listening to the Log 298
17.2 Using the LogService in Toast 298
17.3 Using the LogReaderService 301
17.4 Toast's logutility Class 303
17.5 Equinox's LogService Implementations 304
17.6 Summary 306

Chapter 18 HTTP Support 307
18.1 The HttpService 308
   18.1.1 Concepts 308
18.2 Registering and Unregistering a Servlet 309
18.3 Declarative HTTP Content Registrations 312
18.4 Using Jetty 313
18.5 HTTP Contexts and JAAS Integration 314
   18.5.1 Basic HTTP Authentication and Login 314
   18.5.2 Running the Secured Client 317
18.6 Troubleshooting 318
   18.6.1 bindException 319
18.6.2 On Which Port Is the HttpServletRequest Listening? 319

18.7 Summary 320

Chapter 19 Server Side 321

19.1 Servers and OSGi 322
19.2 Embedding the Back End in a Web Application 323
  19.2.1 Updating the Product 324
  19.2.2 The Web Application Root Files 326
  19.2.3 Building the Web Application 329
  19.2.4 Running the Web Application 330
  19.2.5 Troubleshooting 332
  19.2.6 <init-param>s 333
19.3 Remote Services in OSGi 333
  19.3.1 The Eclipse Communication Framework 334
  19.3.2 Remote Services 335
  19.3.3 Distributed Toast 335
  19.3.4 Remote Service Host 336
  19.3.5 Remote Service Client 338
  19.3.6 Service Discovery 339
  19.3.7 Running the Distributed System 340
19.4 Summary 341

Chapter 20 Release Engineering 343

20.1 What Is PDE Build? 344
20.2 Bundle build.properties 345
  20.2.1 Control Properties 346
  20.2.2 Using Custom Build Scripts 347
20.3 Setting Up a Builder 347
  20.3.1 Tweaking the Target for PDE Build 348
  20.3.2 build.properties 348
20.4 Running the Builder 353
20.5 Tweaking the Build 356
  20.5.1 Customizing the Build Scripts 356
### Contents

20.5.2 Repositories and Additional Dependencies 357  
20.5.3 Fetching from an SCM System 358  
20.5.4 Fetching the Maps 360  
20.5.5 Auto-substitution of Version Numbers 360  
20.5.6 Qualifying Version Numbers 361  
20.5.7 Identifying and Placing Root Files 362  
20.6 Building Add-on Features 363  
20.6.1 Setting Up a Feature Builder 364  
20.6.2 build.properties 364  
20.6.3 Running the Feature Build 365  
20.7 Building WARs 367  
20.8 Summary 367

### Part IV Reference

**Chapter 21 Dynamic Best Practices** 371  
21.1 Dynamism and You 371  
21.2 Dynamic Aspects of Toast 372  
21.3 Dynamic Challenges 374  
21.4 Dynamic Awareness 374  
21.4.1 Object Handling 376  
21.4.2 Bundle Listeners 377  
21.5 The Extender Pattern and bundleTracker 378  
21.6 Dynamic Enablement 379  
21.6.1 Cleaning Up after Yourself 380  
21.7 The Dynamics of Startup and Shutdown 382  
21.7.1 Start Levels 382  
21.7.2 Proper Use of Services 383  
21.7.3 Shutting Down Is Not Always Easy 384  
21.8 Summary 385

**Chapter 22 Integrating Code Libraries** 387  
22.1 JARs as Bundles 388  
22.2 Bundling by Injection 388
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>22.3 Bundling by Wrapping</td>
<td>390</td>
</tr>
<tr>
<td>22.4 Bundling by Reference</td>
<td>392</td>
</tr>
<tr>
<td>22.5 Bundling Using bnd</td>
<td>394</td>
</tr>
<tr>
<td>22.6 Troubleshooting Class Loading Problems</td>
<td>394</td>
</tr>
<tr>
<td>22.6.1 Issues with <code>Class.forName()</code></td>
<td>395</td>
</tr>
<tr>
<td>22.6.2 Issues with Context Class Loaders</td>
<td>399</td>
</tr>
<tr>
<td>22.6.3 Managing JRE Classes</td>
<td>401</td>
</tr>
<tr>
<td>22.6.4 Serialization</td>
<td>402</td>
</tr>
<tr>
<td>22.7 Summary</td>
<td>403</td>
</tr>
<tr>
<td>Chapter 23 Advanced Topics</td>
<td>405</td>
</tr>
<tr>
<td>23.1 The Equinox Console</td>
<td>406</td>
</tr>
<tr>
<td>23.1.1 Extending the Equinox Console</td>
<td>407</td>
</tr>
<tr>
<td>23.2 Roles in OSGi</td>
<td>409</td>
</tr>
<tr>
<td>23.3 The Shape of Bundles</td>
<td>411</td>
</tr>
<tr>
<td>23.4 Fragments</td>
<td>413</td>
</tr>
<tr>
<td>23.5 Singletons</td>
<td>415</td>
</tr>
<tr>
<td>23.6 Bundle Lifecycle</td>
<td>416</td>
</tr>
<tr>
<td>23.6.1 Lifecycle States</td>
<td>416</td>
</tr>
<tr>
<td>23.6.2 <code>BundleActivator</code></td>
<td>418</td>
</tr>
<tr>
<td>23.6.3 The Downside of Activators</td>
<td>418</td>
</tr>
<tr>
<td>23.6.4 Uses for Activators</td>
<td>419</td>
</tr>
<tr>
<td>23.7 Bundle Activation Policy</td>
<td>419</td>
</tr>
<tr>
<td>23.8 Controlling Bundle Start</td>
<td>421</td>
</tr>
<tr>
<td>23.8.1 Persistent Starting</td>
<td>422</td>
</tr>
<tr>
<td>23.8.2 Enabling the Activation Policy</td>
<td>422</td>
</tr>
<tr>
<td>23.8.3 <code>osgi.bundles</code></td>
<td>422</td>
</tr>
<tr>
<td>23.9 Class Loading</td>
<td>423</td>
</tr>
<tr>
<td>23.9.1 Class Lookup Algorithm</td>
<td>424</td>
</tr>
<tr>
<td>23.9.2 Declaring Imports and Exports</td>
<td>424</td>
</tr>
<tr>
<td>23.9.3 Importing versus Requiring</td>
<td>426</td>
</tr>
<tr>
<td>23.9.4 Optionality</td>
<td>426</td>
</tr>
<tr>
<td>23.9.5 The <code>uses</code> Directive</td>
<td>426</td>
</tr>
<tr>
<td>23.9.6 Re-exporting</td>
<td>427</td>
</tr>
</tbody>
</table>
Chapter 24 Declarative Services Reference

24.1 Component XML Schema v1.1.0

24.1.1 Declaring the XML Namespace and Schema
24.1.2 The <component> Element
24.1.3 The <implementation> Element
24.1.4 The <property> Element
24.1.5 The <properties> Element
24.1.6 The <service> Element
24.1.7 The <provide> Element
24.1.8 The <reference> Element

24.2 The Component Lifecycle

24.2.1 Satisfying a Component’s Configuration
24.2.2 Component Activation, Deactivation, and Modification
24.2.3 Accessing Referenced Services
24.2.4 Component Immediacy
24.2.5 Component Properties

24.3 Summary

Index